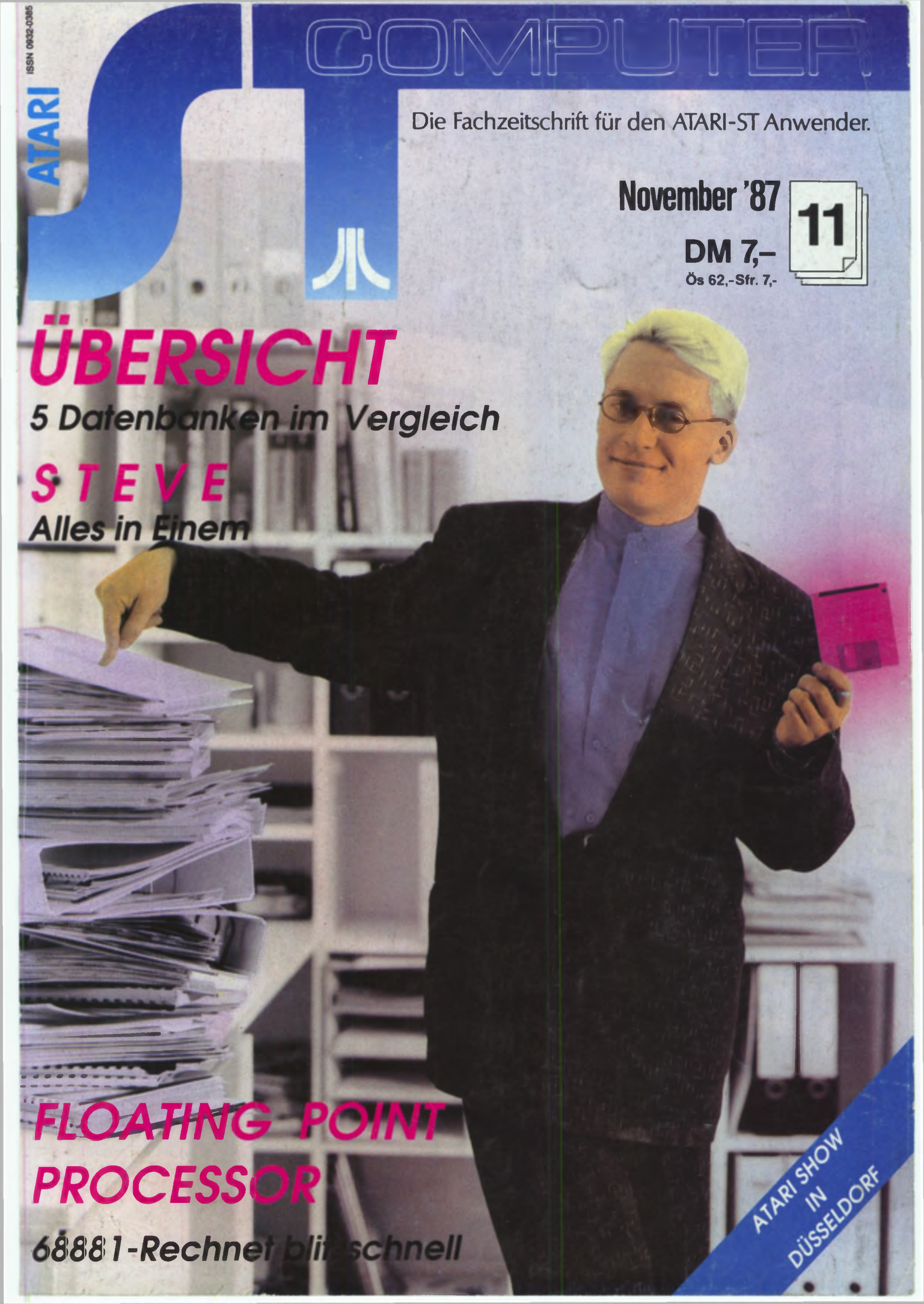
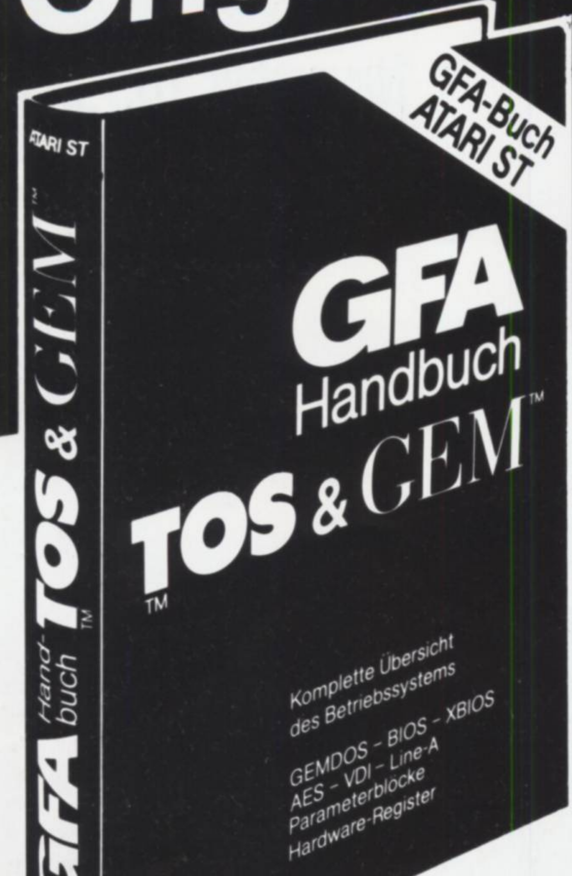
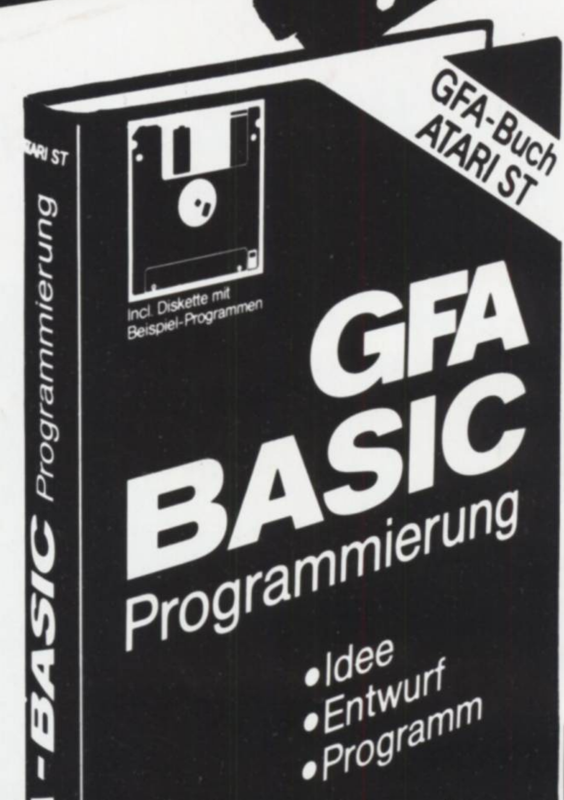

ATARI
ISSN 0932-0385
ST COMPUTER
Die Fachzeitschrift für den ATARI-ST Anwender.
November '87
11
DM 7,–
Ös 62,-Sfr. 7,-
ÜBERSICHT
5 Datenbanken im Vergleich
STEVE
Alles in Einem
FLOATING POINT
PROCESSOR
68881-Rechnet blitzschnell
ATARI SHOW IN DÜSSELDORF

Original GFA-Bücher

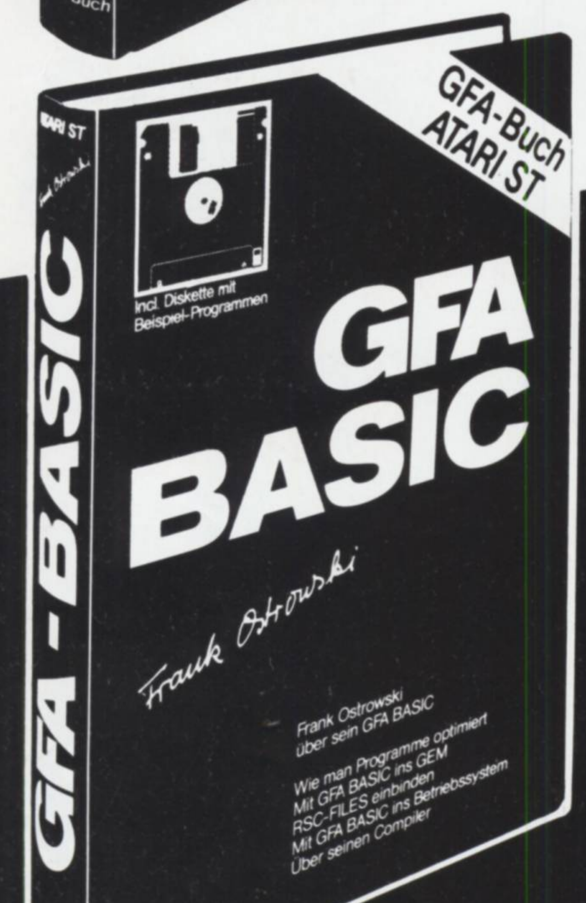


◀ GFA-Handbuch TOS & GEM DM 49,–

neu



◀ GFA-BASIC Programmierung inclusive Diskette DM 49,–



◀ GFA-BASIC-Buch inclusive Diskette DM 79,–

...Anruf genügt: 02 11/58 80 11
GFA-CLUB, GFA-PC-Software bitte Info anfordern

GFA Systemtechnik GmbH

Heerdter Sandberg 30
D-4000 Düsseldorf 11
Telefon 02 11/58 80 11

GFA SYSTEMTECHNIK



Im Zeichen der Messen

Der Weltraum – unendliche Weiten, Myriaden von Sternen, Galaxien entstehen und vergehen wieder. Ein jährlich wiederkehrendes Sternzeichen am Computerhimmel wird astrologisch repräsentiert durch ein großes „M" mit einem Pfeil nach rechts (M->). Im Lexikon der Computeroskopie findet sich dazu folgender Absatz: „...steht das in jüngster Zeit vor allem in Westeuropa und Nordamerika beobachtete Zeichen wahrscheinlich für „Zur Messe geht's da lang" oder „Massenhafte Volksbewegungen in dieser Richtung vermutet". Die unter dem Einfluß dieses Sternzeichen Stehenden treten in der Regel mit wirrem Blick, körperlichen Gebrechen, Atemnot, zerschundenen Füßen und Quetschungen auf. Vereinzelte Exemplare scheinen auch unter Suchterscheinungen zu leiden, denen man in Fachkreisen den Namen „Datenreise-Syndrom" gegeben hat...".

Nach dem Messebesuch in London könnte man diesem Abschnitt noch folgenden Satz hinzufügen: „Die Krankheit breitet sich in Form eines Virus aus und ist unbedingt ansteckend".

Zwei Messen haben wir in den letzten zehn Tagen hinter uns gebracht – die ATARI SHOW in Düsseldorf und die PERSONAL COMPUTER WORLD (PCW) in London. Unter dem direkten Eindruck dieser beiden Messen drängt sich vor allem der grundverschiedene Aufbau dieser beiden Shows auf. Die Unterschiede sind wohl am ehesten durch die gegensätzliche Bedeutung des Atari ST in der Bundesrepublik und in England zu erklären.

In der Bundesrepublik wurde der Atari ST bisher zu rund 80 Prozent mit monochromen Monitoren verkauft. Dies ist ein eindeutiges Zeichen dafür, daß der Rechner hier professionell oder zumindest semiproifessionell genutzt wird. In England ist es genau umgekehrt. Der Atari ST wird dort wegen seines – im Vergleich mit allen anderen 68000er Rechnern – sehr günstigen Preisen fast nur mit Farbmonitor verkauft und hat sich als DER Spielerechner für gehobene Ansprüche durchgesetzt.

Entsprechend ist es auch mit der Software. Die meisten „Inselbewohner" sind, wenn schon Programmierer, dann in der Spieleproduktion tätig. Entsprechend groß ist die Anzahl der britischen Softwarehäuser, die fast nur Spiele anbieten. Gut ist auch die Qualität vieler englischer Spiele, z. B. Airball (MicroDeal) oder Terrorpods (Psygnosis). Die Softwarehäuser Kuma und Metacomco, die sich ausschließlich mit „ernsthaften Anwendungen" befassen, sind im Vereinigten Königreich eher die Ausnahme.

Dies ist auch der Grund, warum es in puncto Anwendersoftware von der PCW fast kaum Neues zu berichten gibt. Die wirklich guten Programme sind in dieser Sparte im Moment sehr häufig „made in Germany", d. h. schon bestens eingeführt. Zu sehen waren in London u. a. Programme wie GFA-Basic, Omikron-Basic, Signum! und Aladin. Deutlichstes Zeichen, wie stark ATARI auf den deutschen Programmierermarkt setzt, war die Zusage von Sig Hartmann (Atari USA) in Düsseldorf, den deutschen Softwarefirmen „...jede nur mögliche Unterstützung..." zu geben, um ihren „...Produkten auch den amerikanischen Markt zu öffnen...".

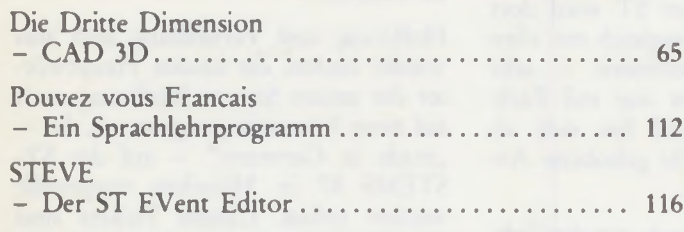
Hoffnung und Verheißung sind also wieder einmal die beiden Hauptwörter der neuen Saison. Hoffnung auch auf neue Anwenderprogramme, die – „made in Germany" – auf der SYSTEMS 87 in München vorgestellt werden sollen. Unsere Tickets sind schon reserviert. Messezeit bedeutet (Daten-)reisezeit.

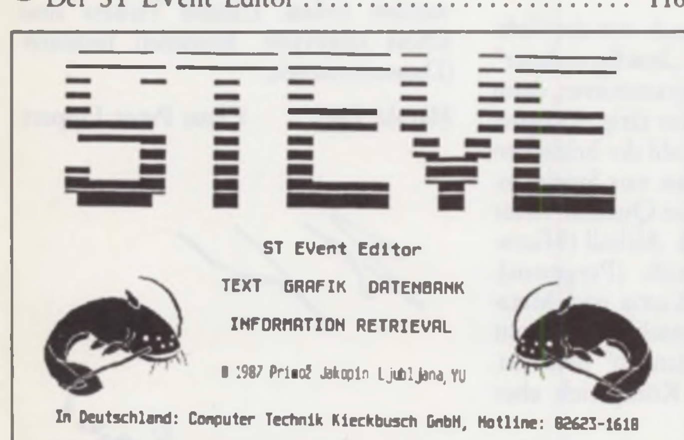
Harald Egel Claus Peter Lippert

Allgemeines

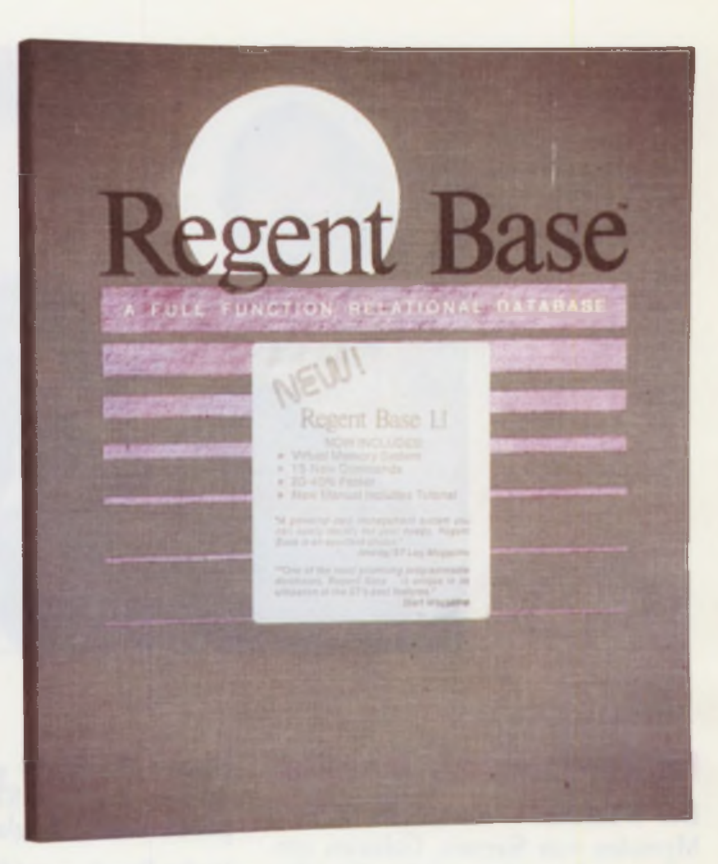
Software







Hardware



Fünf relationale Datenbanken im Vergleich

Das Problem der Speicherung großer Datenmengen und der anschließende schnelle Zugriff auf die Daten war schon immer ein 'heißprogrammiertes' Thema im Softwarebereich. Ein wichtiges Schlüsselwort für die so entstandenen Datenbanken ist „relational". Wir testeten für Sie nun fünf dieser sogenannten Relationalen Datenbanken. Im Ring stehen ADIMENS ST, dBASE II, REGENT BASE, STANDART BASE und SUPERBASE ST. Entscheiden Sie selbst.

Co-Prozessor MC 68881
– Mit Highspeed durch die Register

Wer bis jetzt neidisch auf die IBM-kompatiblen Rechner mit ihren 8087 Mathematik-Co-Prozessoren geblickt hat, dem sei jetzt geholfen. Denn endlich sind die ersten mathematischen Co-Prozessoren für den ST lieferbar. Wir wollen uns diesem Thema in zwei Folgen widmen. Die erste Folge, die Sie in dieser Ausgabe finden, behandelt die Hardware eines solchen Arithmetik-Prozessors.

Programmierpraxis

– eine neue, alte Sparte in der ST

Damit die Tips und Tricks für die Programmierung des ST nicht immer so verstreut in der ST zu finden sind, haben wir jetzt eine neue Sparte eröffnet. Sie nennt sich „Programmierpraxis" und liefert hoffentlich für jedermann viele gute Anregungen und Programme.

ATARI trumpft auf

– Die ATARI SHOW in Düsseldorf

Vom 18.9.–20.9. fand in Düsseldorf die erste ATARI SHOW in Deutschland statt. Fast alles was Rang und Namen hat, war auf dieser Messe vertreten. Gezeigt wurden eine Vielzahl neuer Produkte, die besonders den Anwenderbereich betreffen. Wir laden Sie zu einem kleinen nachträglichen Rundgang über diese ATARI Messe ein.



Programmierpraxis

Grundlagen

Kurse

Aktuelles



Adventure selbst schreiben

Wer die Idee zu einer Adventure-Handlung hat, braucht sich jetzt nicht mehr mit den Einzelheiten der Computer-Programmierung auseinandersetzen, um seine Idee zu realisieren. Dank eines neuartigen Programmes, dem Adventurix-Compiler, wird das Programmieren eines Adventures zu einer 'Spielerei'. Die Beschreibung des Adventures wird als ASCII-File eingegeben und durch den Compiler in ein lauffähiges Programm umgesetzt.

Das Programm erlaubt die Erstellung von Spielen in jeder Sprache und jeder Bildschirmauflösung. Bei der Anordnung der Räume sind der Phantasie keine Grenzen gesetzt. Mehrdimensionale Labyrinthe sind ohne weiteres möglich.

Detlef Pleiß
Modemannskamp 5
4500 Osnabrück

Aktienverwaltung am ST

Von dem in Braunschweig ansässigen Softwarehaus Partner-Software wird ein Programm angeboten, das die Verwaltung von Aktien vereinfacht. Das Programm Aktien-Verwaltung/Analyse verwaltet bis zu 240 Aktien, unterteilt gleichzeitig in vier Märkte und erstellt Chart-Graphiken und Analysen.

Das Programm ist in GEM eingebunden und wird komplett mit der Maus gesteuert. Aktien-Verwaltung wird mit einer ausführlichen Bedienungsanleitung geliefert und kostet DM 298,–

Partner Software
Klaus Tappe
Virchowstr. 31
3300 Braunschweig
Tel.: 05 31 / 89 40 29

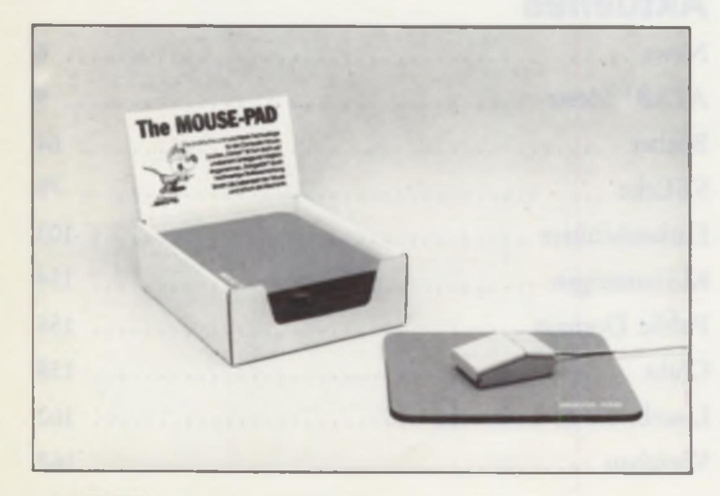
Mouse-Pad

Mouse-Pad ist eine rutschfeste Tischauflage für Computer-Mäuse. Das Format beträgt 270 x 220 mm und ist aus festem Neopren-Schaumstoff mit einer antistatischen Textilbeschichtung aufgebaut. Eine solche Tischauflage bildet eine bessere Auflage als die normale Tischfläche. Der Preis für ein Mouse-Pad beträgt DM 19,80

Metra-Sound
Bayenthalgürtel 45
5000 Köln 51

Blitter For All But Chips Not Free

Um allen ATARI ST Besitzern den Blitter zugänglich zu machen, wird dieser Baustein jetzt zum Nachrüsten in einem Dual-in-Line Gehäuse angeboten, so daß er direkt und problemlos auf den 68000er Prozessor gelötet werden kann. Der Blitter wird allerdings nur von der neuesten TOS-Version erkannt und angesprochen. Besitzer eines alten ST-Modells werden dann auch die neue Version des TOS (Blitter-TOS) erwerben müssen, um den Blitter nutzen zu können. Der Preis stand zum Redaktionsschluß noch nicht fest.





Persönlicher Fernkopierer NEFAX-10

Ein Telefax, trotz seiner unbestrittenen Hilfe bei der schnellen Datenübertragung von Dokumenten, ist wegen seinen großen Abmessungen ein sehr unhandliches Gerät.

NEC Deutschland bietet jetzt einen Fernkopierer an, der wegen seiner geringen Proportionen auf jeden Schreibtisch paßt. Dabei bietet das kompakte und transportable Gerät ein Leistungsspektrum, das sonst nur bei größeren Fernkopierern zu finden ist.

Das Gerät ist in der Lage wahlweise automatisch oder manuell zu empfangen und zu senden. Durch eine automatische Kontrastanpassung, die den Hintergrund und die Druckqualität des zu sendenden Dokuments berücksichtigt, werden auch schwach gedruckte Vorlagen kontrastreich und fein aufgelöst wiedergegeben.

Die Multisync Familie wird erweitert

Zwei weitere Farbmonitoren der MultiSync-Technologie wurden von NEC vorgestellt.

Der MultiSync Plus ist ein erweitertes Modell des alten Multi-Sync. Das Frequenzspektrum wurde fast verdoppelt (21,8 – 45 kHz), die Auflösung beträgt jetzt 960x720 (beim alten MultiSync nur 640x480) und die Verbindungsmöglichkeiten wurden ebenso erweitert.

Der MultiSync XL ist der große Bruder aller drei Modelle. Der Neuling der Multi-Familie besitzt eine Auflösung von 1024x768 Pixel und eine sehr hohe Horizontal- bzw. Vertikal-Frequenz. Dieses Modell ist besonders für den Einsatz im CAD-Bereich vorgesehen.

NEC Deutschland GmbH
Klausenburger Straße 4
8000 München 80

Floppy-Speeder hilft dem ATARI beim Lesen und Schreiben

Eine reine Softwarelösung, die den Datentransfer zwischen Rechner und Lauf-

BILLIG-SPIELE FÜR AMIGA und ST MÜSSEN NICHT »BILLIG« SEIN!



CITY DEFENCE

von Uwe Kühner

Eine schier endlose Zahl tödlicher Laserstrahlen aus dem Weltall kann nur gestoppt werden, wenn Sie zum richtigen Zeitpunkt Ihre Bodenraketen abschiessen, bevor die Städte zerstört werden. 1/2 Spieler; Maussteuerung. **Lieferbar für AMIGA**





GOKART RACING

von Anco

Ein exzellentes Autorennen, bei dem ein echter Champion gefordert wird. Bei unterschiedlichsten Witterungsbedingungen und Strecken müssen Sie vorher Reifen und Übersetzung wählen. 8 Kurse, 1/2 Spieler (gleichzeitig); Joysticksteuerung.
Lieferbar für AMIGA und ST



STRIP POKER

von Artworx

Verbringen Sie einen schönen Abend mit einer Runde Strip Poker. Mit Susi und Melissa stehen Ihnen 2 bildschöne, aber spielstarke Partnerinnen zur Verfügung. Bequeme Spielsteuerung mit der Maus. **Lieferbar für AMIGA und ST**







EMERALD MINE

von Klaus Heinz und Volker Wertich

Ein Riesen-Arcade-Adventure mit über 100 Leveln und 2-Spieler-Option (gleichzeitig im Teamwork!); Joysticksteuerung. *"Die flotte Grafik mit witzigen Monster-Sprites und vor allem der Sound gefallen mir sehr gut: Die fetzige Titelmusik und die digitalisierten Sound-Effekte sind exzellent. Das Verblüffendste an diesem tollen Spiel ist aber der sehr niedrige Preis: Wer hier nicht zuschlägt, ist selber schuld." (HAPPY COMPUTER)*

Lieferbar für AMIGA und ST



FORTRESS UNDERGROUND

von Andreas von Lepel

In einer gigantischen unterirdischen Höhle (640 Bildschirme groß) müssen Sie ein feindliches Kraftwerk aufspüren und eliminieren. Aber der Weg dorthin ist lang und gefährlich. Joysticksteuerung.
Lieferbar für AMIGA



Lieferbar für AMIGA



Lieferbar für AMIGA



Lieferbar für AMIGA



Lieferbar für AMIGA/ST



Lieferbar für ST



Lieferbar für AMIGA



Lieferbar für AMIGA



Lieferbar für ST





Lieferbar für ST



Lieferbar für AMIGA/ST

SENSATIONELL

je Spiel nur DM

29.95

unverb. Preisempfehlung

KINGSOFT-Spiele gibt's überall da, wo es gute Software gibt.

SPITZEN-SOFTWARE ...natürlich von

KINGSOFT

MADE IN GERMANY

F. Schäfer · Schnackebusch 4 · 5106 Roetgen
☎ 02408/5119 (nicht aufgeben!) Fax 02408-5213

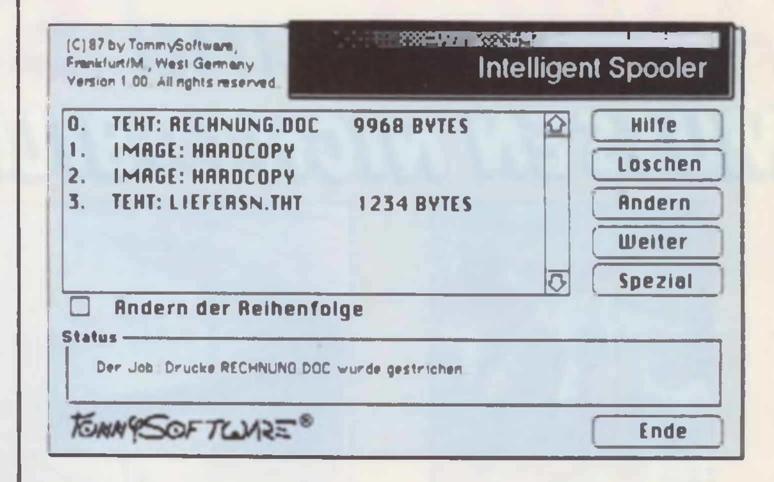




Bei uns gibt's jeden Monat Neuerscheinungen für Ihren AMIGA und ST - am besten kostenlos den aktuellen Katalog anfordern!

Alle Spiele selbstverständlich mit ausführlicher deutscher Anleitung, Super-Grafik und fantastischen Digisound-Effekten.

News



werk beschleunigt, bietet das Frankfurter Softwarehaus 'TommySoftware'. Der durchschnittliche Zugriff beim Floppybetrieb wird durch Cache-Technik um den Faktor 10 erhöht. Bei der Harddisk wird eine Geschwindigkeitssteigerung um etwa den Faktor 2 erreicht. Der Floppy-Speeder, der mit einer deutschen Anleitung geliefert wird, kostet DM 98,–.

Noch zwei weitere Produkte werden vom gleichen Unternehmen für den ATARI ST angeboten. Freezer ist ein Kopierprogramm, womit Daten wahlweise komprimiert oder unkomprimiert auf Diskette abgelegt werden können. Das Programm, das in allen Auflösungen läuft und 1 MB RAM erfordert, kostet DM 148,–.

Bei dem anderen Programm handelt es sich um einen intelligenten Spooler, der in der Lage ist, bis zu 16 MB auf einer Harddisk zu puffern, und dann selbständig die Daten auf einen Drucker zu senden. Der intelligente Spooler, der selbst nur 60 KB RAM beansprucht, arbeitet mit Precatch-Logik, so daß kurze Texte direkt 'gespoolt', längere zuerst auf Platte abgelegt werden und von dort später selbständig wieder geladen werden. Der Preis für den intelligenten Spooler beträgt DM 98,–.

TommySoftware
Gutzkowstraße 35
6000 Frankfurt/M. 70
Tel.: 069/614046

Wehrdat. Eine juristische Datenbank für Adimens ST

Gerichtliche Entscheidungen fallen jeden Tag an. Je nach Rechtslage ist die juristische Vorbereitung dementsprechend komplex. Um die Arbeit auf den Gebieten des Wehrpflichts-, Kriegsdienstverweigerungs- und Zivildienstrechts zu erleichtern, wird jetzt auf zwei Disketten eine Datenbank namens WEHRDAT angeboten, die unter ADIMENS ST arbeitet. Der Preis für WEHRDAT beträgt DM 399,–. Es besteht aber auch die Möglichkeit die Datenbank zusammen mit ADIMENS ST für DM 598,– zu erwerben.

Werner Forkel
Kriegsstr. 29
7500 Karlsruhe 1
Tel.: 0721/377786

Aus zwei mach drei

Eine Erweiterung, die den Anschluß von drei Laufwerken ermöglicht, bietet die Firma Miwiko Computertechnik an. Zwischen dem zweiten und dritten Laufwerk kann mittels einem Schalter problemlos umgeschaltet werden, ohne daß ein Reset erfolgt. Zwei Leuchtdioden zeigen ständig an, welches Laufwerk gerade aktiviert ist. Die Floppybox (so heißt diese Erweiterung) kostet DM 99,–.

Miwiko Computertechnik
Tel.: 0221/382020



Neues OMIKRON.Basic Handbuch

Zum OMIKRON.Basic Interpreter gibt es ab sofort ein neues, überarbeitetes Handbuch. Hinzugekommen sind einige erweiterte Beispiele und eine Übersicht über alle BIOS, GEMDOS und XBIOS-Befehle mit Parameterangabe und deren Funktion. Weiterhin sind alle GEM-Befehle aufgelistet und deren prinzipielle Einbindung ins Basic beschrieben. Somit ist auch den Einsteigern beim Umgang mit GEM geholfen.

Besitzer des Basic-Interpreters, und nur diese, können bei Interesse das neue Handbuch für DM 30,– direkt bei OMIKRON.Software beziehen. Nähere Informationen bei:

OMIKRON.Software
Erlachstraße 15
7534 Birkenfeld 2
Tel.: 07082/5386

18. - 20. September 1987

ATARI® messe

Messehalle 1 · Messegelände · Düsseldorf

Die erste internationale ATARI-Messe

Eine eigene Messe zu veranstalten, beweist, daß ATARI es geschafft hat, innerhalb von nur zwei Jahren ein großes Publikum, vor allem mit der ST-Generation, anzusprechen. Kein Wunder, daß diese Show für ATARI ein Erfolg war, aber auch für den Besucher, der, wie nie zuvor, geballte Informationen rund um ATARI in einer einzigen Halle finden konnte. Für all diejenigen, die keine Möglichkeit hatten, in Düsseldorf dabei zu sein, wollen wir hier das Wichtigste zusammenfassen.

Vom 18. bis 20. September fand auf dem Düsseldorfer Messegelände die erste internationale ATARI-Messe Deutschlands statt. Fast 20 000 Besucher nutzten die Gelegenheit, sich in der Halle 1 einen umfassenden Überblick über all das zu verschaffen, was derzeit rund um die ATARI Computer geboten wird. Obwohl ATARI die gesamte Produktpalette zeigte, also auch den 800 XE und den neuen PC, waren solche um die ST Serie wohl die Anziehungspunkte Nummer eins.

Aber nicht nur für die Besucher, sondern auch für die etwa 80 Aussteller war die Messe, bedingt durch die Möglichkeit, ihre Produkte nicht nur vorzuführen, sondern auch an Ort und Stelle verkaufen zu können, sicherlich ein Erfolg.

Parallel zum „Messegedränge" fanden täglich sieben Workshops statt, die dem Besucher die Möglichkeit gaben, sich über aktuelle Themen, wie Desktop Publishing, relationale Datenbanken oder MIDI-Software zu informieren. Besonders stark frequentiert wurde der Vortrag von Shiraz Shivji, ATARIs Chefentwickler und Vater der ST-Modelle. Sein Vortrag über „Future Developments" gewährte einen kleinen Einblick in das, was künftig aus den ATARI-Entwicklungslabors zu erwarten ist.

Entwicklungen der Zukunft – Shiraz Shivji berichtet über den Stand der Dinge

Über zukünftige Planungen der Firma ATARI gab es viele Gerüchte, angefangen vom leistungsstärkeren Modell mit 68020 Prozessor, oder dem **CD-ROM-Laufwerk**, über einen neuen geheimnisvollen Soundchip, bis hin zu einem ST-Modell mit höherer Grafikauflösung. Schon der überfüllte Saal bei der Rede von Herrn Shivji, ließ die Wichtigkeit des Themas erkennen. Und tatsächlich konnte man endlich einmal einige offizielle Angaben über Zukunftsprojekte erfahren!

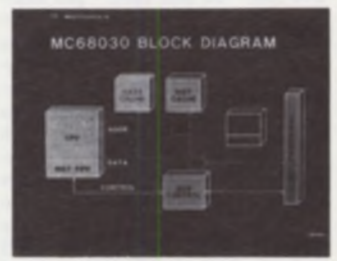
Beginnen wir mit dem CD-ROM-Laufwerk, denn dieses soll noch in diesem Jahr für etwa DM 1000,– zu haben sein. Ein CD-ROM-Laufwerk ist ein optischer Massenspeicher mit riesiger Kapazität, von dem aber nur gelesen werden kann, d. h. der Anwender ist auf das Angebot an fertig bespielten Disketten (CDs) der Hersteller angewiesen. Der ATARI CD-ROM-Player wird auch in der Lage sein, gewöhnliche CDs der Audiotechnik über die Stereoanlage abzuspielen. Man darf gespannt sein, welche „Daten" auf solchen optischen Platten angeboten werden.

Höhere Auflösung mit dem EST

Schon oft hörte man Gerüchte über eine neue Grafikkarte für den ST oder einen Rechner mit höherer Bildschirmauflösung, doch jetzt dürfte es wohl spruchreif sein, zur CeBit im März 1988 soll der Extended ST, der EST, erscheinen. Dieser neue Rechner wird als Prozessor auch den 68000 von Motorola besitzen, jedoch wird der EST in Sachen Grafik den bisherigen ST-Modellen weit überlegen sein. Im Schwarzweißmodus besitzt der EST eine Auflösung von 1280★960 Bildpunkten und im Farbmodus wird er 256 Farben aus einer Palette von 4096 Farben gleichzeitig darstellen können. Mit dieser fast um das Fünffache verbesserten Auflösung wird sich der EST besonders für DTP- und CAD-Anwendungen eignen.

Superrechner mit 68030 Prozessor

Nachdem die Spannung auf neue Rechner-Projekte hoffentlich nicht mehr zu steigern ist, wollen wir auch darüber etwas verraten. Öfter schon war eine „Blackbox" mit einem 68020-Prozessor im Gespräch, dann wurde gemunkelt, es würde gleich ein 68030 eingebaut. Tatsächlich scheint man sich bei ATARI nun für eine Art Workstation, die einen **68030-Prozessor** besitzen soll, entschieden zu haben. Die Mega-ST's werden für dieses System als Bildschirm-Station verwendet. ATARI möchte mit diesem neuen Superrechner gleich einen ganzen Schritt weitergehen und einen Rechner bauen, der erheblich schneller sein soll, als bisher gängige 32-Bit Rechner. Die Rechenleistung des 68030 beträgt etwa 4 MIPS (Millionen Instruktionen pro Sekunde). Das Betriebssystem wird UNIX V sein, vorausgesetzt Motorola kann die dafür notwendige PMMU bis dahin in einer fehlerfreien Version liefern. Ferner sollen eine GEM ähnliche Benutzeroberfläche (X-Window) und ein Netzwerksystem bereits mitgeliefert werden. Als Preisregion für dieses Modell wurden DM 6000,– und mehr angegeben. Mehr Details waren noch nicht zu erfahren.



Transputer-Rechner

Das mit Abstand spannendste Projekt, das Herr Shivji vorstellte, geht aber noch weit über die 68030-Workstation hinaus. Das ATARI-Team kam gerade aus Chambridge, wo der Prototyp eines Rechners auf Transputerbasis besichtigt wurde (Projekt-Spitzname: Mini-Cray). Dieses Gerät soll modular aufgebaut sein, bis zu 30 Transputerkarten aufnehmen können und über eigene Grafik verfügen. Als Transputer soll der INMOS T-800 Chip (Taktfrequenz 20 MHz) verwendet werden. Dieser Prozessor ist ein 32-Bit Typ, der über eine Hardware-FPU (Floating Point Unit) verfügt, die bis zu 1,5 Millionen Floating-Point Operationen pro Sekunde (MFLOPS) leistet. Wie gesagt, man kann dem Grundsystem, das wie die 68030-Workstation den Mega-ST als Terminal nutzen soll, nach Wunsch und Geldbeutel Transputer hinzufügen. Ein spezielles UNIX-ähnliches, mehrprozessorfähiges Betriebssystem wird an der Universität von Cambridge bereits entwickelt, ebenso die notwendigen Programmierwerkzeuge (über Transputer konnten Sie im vorletzten Heft näheres erfahren).

Die Grafikkarte wird in der größten Version bis zu 32 Bit pro Bildpunkt verarbeiten. Davon dienen 24 Bit für die Farbinformationen, was 16,7 Millionen darstellbare Farbtöne bedeutet. 1 Bit ist reserviert, während die restlichen 7 Bits dem Anwender für Zusatzinformationen zur Verfügung stehen.

Besonders interessant wird dieser Rechner, dessen maximale Rechenleistung um die 300 MIPS liegen dürfte, vor allem für wissenschaftliche Aufgaben und den Bildverarbeitungs- und Grafikbereich sein.

Doch kommen wir nach so vielen MIPS, MFLOPS und Computerleistung wieder auf den Boden der Realität zurück und berichten über das Wichtigste, das es bereits zu sehen gab.

Die neue 20 MByte Festplatte

Ab sofort liefert ATARI nur noch die neue Festplatte mit der Bezeichnung **SH 205.** In den Abmessungen und im Design ist sie den Mega ST-Modellen angepaßt. Aber nicht nur das Gehäuse, sondern auch das eingebaute Schaltnetzteil entspricht dem der Mega STs, leider jedoch nicht das Laufgeräusch des eingebauten Lüfters, der um einiges lauter ist. In der SH 205 wird ein 3,5 Zoll Plattenlaufwerk verwendet, so daß eigentlich noch Platz für ein zweites Platten- oder auch Floppy-Laufwerk vorhanden ist. Vielleicht wird es ja so etwas von ATARI oder einem anderen Hersteller als Nachrüstsatz geben? Technisch neu ist das Vorhandensein einer DMA-IN **und** einer DMA-OUT Buchse. Dadurch ist es erstmals möglich, mehrere Festplatten – um genau zu sein – maximal acht Festplatten gleichzeitig an den ST anzuschließen. Über den DMA-OUT

der Festplatte kann dann auch der neue ATARI Laserdrucker SLM 804 angeschlossen werden. Zusammen mit der Festplatte wird auch ein neuer Harddisk-Treiber ausgeliefert, der das Booten von der Platte ermöglicht. Erfreulich ist, daß trotz dieser Neuerungen der Preis von DM 1298,– der gleiche geblieben ist, wie beim Vorgängermodell, der SH 204.



Die neue SH 205 mit DMA in und DMA OUT

Neue und interessante Produkte, die nicht aus dem Hause ATARI stammen

Desktop Publishing – ein Muß für jedermann?

Firmen, die Desktop Publishing (DTP) Produkte herstellen, sprießen wie Pilze aus dem Boden. Keine Firma kann zur Zeit ein fertiges und ausgereiftes Programm vorstellen. So waren auch in Düsseldorf nur Vorversionen zu sehen. Allerdings scheint das DTP Programm Calamus der Firma DMC (wir berichteten bereits darüber) wirklich professionellen Ansprüchen Rechnung zu tragen und sich auch kurz vor der Fertigstellung zu befinden. Auch beim GFA Publisher gibt es Verzögerungen, er soll aber noch im November ausgeliefert werden. Neu dabei ist auch die Firma Data Becker, die mit einem eigenen Programm ihren Beitrag zum Thema DTP lieferte.

Auch aus England werden uns in naher Zukunft in deutscher Sprache DTP Programme beschert. Zu diesen gehört der, immer noch nicht fertiggestellte, Fleet Street Editor von Mirrorsoft und Calligrapher von Computer Concepts. Ferner kennen wir noch die eine oder andere Firma, die ihr DTP Produkt erst im Frühjahr auf den deutschen Markt bringt. Bleibt die Frage, wieviele ATARI Anwender wirklich ein DTP Programm brauchen, um damit Texte bzw. Grafiken zu gestalten. Für das Layout einer Vereinszeitung oder das Entwerfen eines Prospektes ist ein solches Programm sicherlich hilfreich, zum Schreiben von Briefen oder wissenschaftlichen Texten, auch mit Bildern und Spaltensatz, gibt es schon gute Programme, die nicht das „Schlagwort" Desktop Publishing benutzen.

Der ATARI am Netz

Fast zwei Jahre lang gab es keine Möglichkeit, mehrere ST-Rechner miteinander zu koppeln, dann geht plötzlich der Rummel um Netzwerke so richtig los. Ob dies etwas mit einem endgültigen Durchbruch der ATARI ST Modelle auf dem professionellen Markt zu tun hat? Zumindest sind die Vorteile einer Vernetzung mehrerer ST-Rechner für klein- und mittelständige Betriebe eindeutig. So kann unter Umständen sogar Geld gespart werden, wenn z. B. für mehrere Rechner nur ein Drucker oder nur ein Massenspeicher (Festplatte) gekauft werden muß.

Auf der ATARI Show waren gleich sieben Firmen vertreten, die Netzwerke für den ST anboten. Das preiswerte A-NET der Firma DM Computer GmbH hatten wir bereits im letzten Heft vorgestellt. Ferner zeigten folgende Firmen ihre Produkte zu diesem Thema: Biodata 6272 Niedernhausen, BNT Computerfachhandel 7000 Stuttgart, gdat GmbH 4800 Bielefeld, GTI GmbH 1000 Berlin 45, PAM Software 6500 Mainz und Marvin AG CH-8050 Zürich. Die Preisklasse reicht von knapp DM 500,– bis DM 2000,– pro Rechner, und bei manchen Systemen kommt noch ein einmaliger Betrag von bis zu DM 8000,– hinzu. Ein direkter Vergleich dieser Netzwerke untereinander, ist zur Zeit noch nicht möglich, da auch hier die meisten Systeme noch nicht ausgereift oder viele der vorgesehenen Features noch nicht installiert sind. Momentan müssen wir noch ein wenig warten bis einige Netzwerke lieferbar sind. Übrigens wird ATARI auch mit einem eigenen Netzwerk auf den Markt kommen.

Video Interface und HF-Modulator

Ein neues Videointerface ermöglicht den Anschluß jedes Fernsehers, Farb- und Monochrom-Monitors, sowie jedes Videorecorders an den ATARI ST. Zwei Versionen werden angeboten: V001-ST erzeugt ein PAL (FBAS) Signal und ermöglicht die Farbwiedergabe über gewöhnliche Monitore oder Fernseher mit entsprechendem Eingang. Preis etwa DM 210,– . Die Version V002-ST besitzt zusätzlich einen HF-Modulator, so daß die Wiedergabe des Computerbildes über den Antennen-Eingang eines jeden Fernsehers (Kanal 3) erfolgen kann. Preis ca. DM 250,–.

Minitec-Electronic
Kollaustr. 41-63
2000 Hamburg 54
Tel.: 0 40 / 52 02 61-64

SAM, ein Echtzeit-Videodigitizer

SAM (Silicon Animation Machine) heißt ein neuer Schwarzweiß-Videodigitizer mit besonderen Eigenschaften. Einzigartig ist die Geschwindigkeit, mit der 16 Graustufen digitalisiert werden. 50 Bilder pro Sekunde werden auf dem Monitor wiedergegeben! SAM arbeitet nur in der niedrigsten Auflösung mit 320 x 200 Bildpunkten. Obwohl sich eine solche Auflösung nicht überragend anhört, wirken die digitalisierten Bilder, bedingt durch die relativ hohe Anzahl von Graustufen (16), erstaunlich realistisch. SAM wird am ROM-Port angeschlossen. Benötigt wird ein Rechner mit mindestens 512 KByte freiem Speicher und einem Farbmonitor. SAM kostet etwa DM 980,–.



SAM – schneller Digitizer mit 16 Graustufen

ROTA Datenverarbeitung GmbH
Gochteiner Str. 7
4060 Viersen 1
Tel.: 0 21 62 / 3 10 12



Abdruck eines mit SAM digitalisierten Bildes

Business-Software von Bavaria-Soft

Mit **BSS-PLUS** wird eine völlig neue Software Philosophie, die „Variodynamische Modulsystemsoftware" geboren. Üblicherweise gibt es für verschiedene Bereiche oder auch Branchen, speziell darauf zugeschnittene Programme. Mit BSS-PLUS können Sie durch das „Baukastensystem" die benötigte Software Ihren individuellen Wünschen anpassen. Man erwirbt ein Basissystem, welches dann nach und nach den eigenen Erfordernissen entsprechend erweitert werden kann. Das Basissystem BSS-PLUS kostet DM 299,– und die Erweiterungsmodule – je nach Applikation – zwischen DM 49,– und DM 249,–. Die Fertigstellung der Basis-Applikationen, die eigentlich schon für diesen Sommer angekündigt worden waren, wird nun im Oktober, rechtzeitig zur Systems, erreicht sein. Weitere Module für Ärzte, Zahnärzte, Steuerberater, Rechtsanwälte, Finanzen ect. folgen dann 1988.

Bavaria-Soft Datentechnik GmbH
Otto-Hahn-Str. 25
8012 Ottobrunn bei München
Tel.: 0 89 / 609 78 38

Digitales Oszilloskop

Die schweizer Firma Marvin AG bietet dem engagierten Elektroniker einen sogenannten **Transientenrekorder** an. Dieses Meßgerät verwandelt den ST in ein digitales Speicher-Oszilloskop oder auch in einen Logik-Analysator. Das Gerät kann durch Modul-Technik beliebig ausgebaut werden. Maximal sind bis zu 15 Kanäle bei einer Auflösung von 8-Bit (Vertikale) und einer Samplerate von 20 Millionen Samples (MS) pro Sekunde möglich. Diese Samplerate läßt sich durch Parallelschalten zweier Kanäle noch verdoppeln. Jeder dieser max. 15 Analog-Kanäle kann als Logik-Analysator acht digitale Signale aufzeichnen. Das Gerät, das durch seine technischen Daten durchaus für den professionellen Elektroniker/Entwickler geeignet ist, wird auch einen Preis um die DM 3000,– haben. Im Vergleich zu eigenständigen ähnlichen Geräten ist der Transientenrekorder für den ST eher als preiswert einzustufen. Das Gerät wird erst im Frühjahr '88 auf den deutschen Markt kommen.

Marvin AG
Pries-Str. 23
CH-8050 Zürich

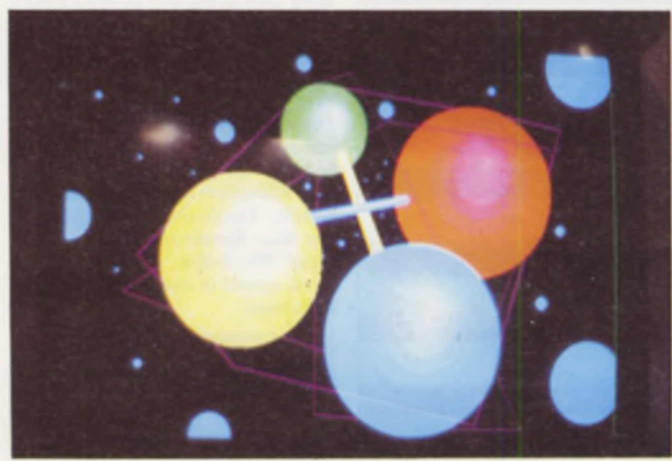


Die neue Produktpalette BSS PLUS

256 Farben gleichzeitig und noch mehr ...

COLOR CAD ist ein schnelles Grafiksystem mit einer Geschwindigkeit von 4 Millionen Pixeln pro Sekunde, einer frei programmierbaren Auflösung von 1024 x 512 Bildpunkten, wovon 820 x 512 gleichzeitig dargestellt werden können. Mit dieser neuen Grafikerweiterung können bis zu 256 Farben auf einem entsprechend guten Farbmonitor dargestellt werden (jeder Multisync-Monitor ist anschließbar). Die Darstellung ist vollkommen flimmerfrei, da die Bildwiederholfrequenz nicht 50 Hz sondern 70 Hz beträgt und nicht im Interlace-Modus gearbeitet wird. Die Programmierung dieses CAD-Systems gestaltet sich sehr einfach. Für fast zehn gängige Programmiersprachen/Systeme existieren bereits Treiber. Vorhandene Programme müssen angepaßt werden, damit ein Betrieb über die COLOR CAD-Karte möglich ist. Die Hardware wird an den ROM-Port und natürlich an der Monitorbuchse des ST angeschlossen. Auf der Karte befinden sich ein eigener Grafikprozessor (HD 63484) und ein 512 KByte großer Bildschirmspeicher. COLOR CAD kostet DM 2700.

Omega Datentechnik
U. Breuer & A. Benninghoff GbR
Sonnenstr. 24
5800 Hagen
Tel.: 0 23 31 / 40 69 73



Ein Grafikbeispiel mit COLOR CAD

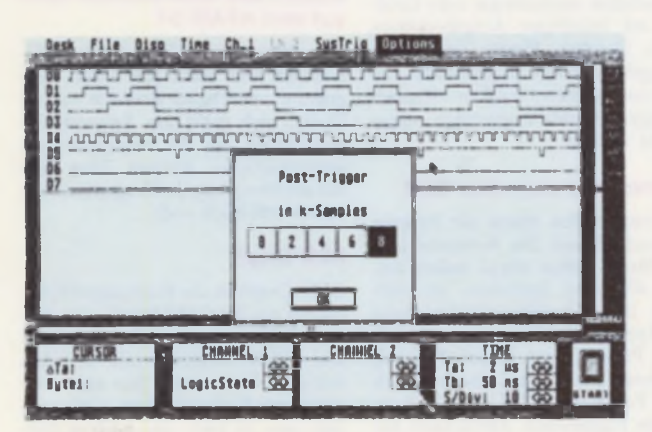
Neues von Application Systems /// Heidelberg

Kurz vor der Markteinführung stehen ebenfalls einige Produkte von Application Systems. So ist demnächst mit einer neuen **Megamax C Version 2.0** zu rechnen, ebenso wird die **Version 2.0 des bekannten Textverarbeitungsprogrammes Signum** bald erscheinen. Als neue Funktionen werden sich Grafiken und Bilder mit Text mischen lassen, und mehrspaltiger Blocksatz wird möglich sein. Weiterhin ist noch in diesem Jahr das in Deutschland entwickelte **Megamax Modula 2** erhältlich. Doch auch im Grafikbereich tut sich einiges. Hier ist mit dem Grafikprogramm **Creator** und dem **Grafik-Compiler Imagic** zu rechnen. Über Imagic wollten wir ja bereits in der ST Computer 7/8 berichten. Leider mußten dann einige Teile des Compilers überarbeitet werden, so daß der Bericht nicht veröffentlicht werden konnte. Imagic ist ein Pascal-ähnlicher Compiler, mit dem u. a. Trickfilme oder auch Produktdemonstrationen erstellt werden können. Als besonderer Reiz lassen sich auch mehrere ST's über die MIDI-Schnittstelle synchronisieren.

Application Systems /// Heidelberg
Englerstr. 3
6900 Heidelberg
Tel.: 0 62 21 / 30 00 02



Die Hardware von COLOR CAD



Digitales Oszilloscop von Marvin AG







Neues aus Heidelberg

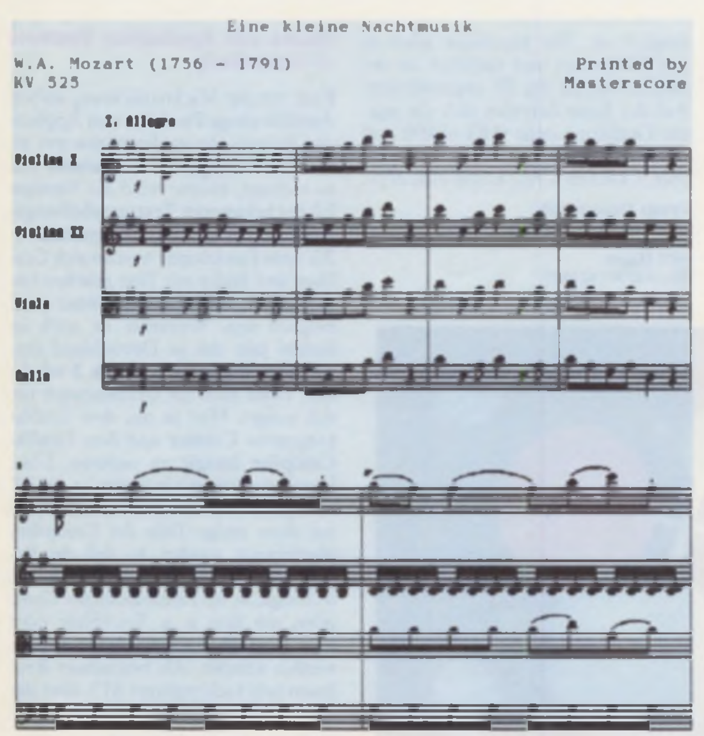
Midi-News

Ungewöhnlich stark waren in Düsseldorf die Musiksoftwarehersteller vertreten.

Steinberg führte die **Version 2.1 des Sequencers Twenty-Four** vor, der sich von früheren Versionen durch einen verbesserten Noten-Editor und einige zusätzliche Funktionen unterscheidet, besonders durch ein 'Logical Edit' genanntes Feature, das die Veränderung von Noten erlaubt, die ganz bestimmten Bedingungen genügen (z. B. alle Noten, die höher als c1 sind, aber nicht länger als eine 8tel-Note).

Das **Notendruckprogramm Masterscore** ist inzwischen auch in einer ersten Version auf dem Markt. Partituren und Stimmen, die sich in einfachen Taktarten bewegen, lassen sich damit in sehr guter Qualität ausdrucken. Für kompliziertere Taktarten und korrekte Vokal-Stimmen fehlt noch die Möglichkeit, Balken per Hand löschen und setzen zu können. Auch die Darstellung polyphoner Vorgänge in einem System ist noch nicht fehlerfrei. Die Eingabe von Text ist an beliebigen Stellen möglich, ebenso von Sonderzeichen. Schlüssel- oder Tonartänderungen sind kein Problem. Auch der Takt kann beliebig gewechselt werden. Auftakte sind möglich, die wichtigsten Sonderzeichen stehen zur Verfügung. Die Eingabe der Stücke sollte möglichst mit dem Twenty-Four erfolgen, ein eigener Editor ist zwar vorhanden und auch recht flott, aber längere Eingaben dürften damit zur Qual werden.

Zusammenfassend läßt sich über dieses Programm sagen, daß alles das, was sich automatisch und nach Regeln erledigen läßt (Aufteilung der Noten im Takt, Balkensetzung in einfachen Taktarten, Verteilung der Systeme auf der Seite), sehr sauber und korrekt dargestellt und gedruckt wird. Auch die Druckgeschwindigkeit ist O.K. Lediglich die manuellen Beeinflussungsmöglichkeiten, die für sehr komplexe Musik



Ein Beispielausdruck von Steinbergs Masterscore im 9-Nadel-Modus

oder gar neue Musik und Vokal-Noten notwendig sind, müssen noch verbessert werden. Ein Druckertreiber für Laserdrucker ist angekündigt.

Ansonsten gab es auf dem Steinberg-Stand eine Reihe von Editoren für verschiedene Midi-Instrumente, sowie das Gehörbildungs-Trainingsprogramm 'Das Ohr', das mehr unter die Rubrik 'Educational Software' fällt.

Eine Neuheit, die nicht gezeigt wurde, sei hier noch angekündigt: Für den sensationellen Preis von DM 790,– wird es ein SMPTE-Interface für den Twenty-Four geben. Das Gerät steckt in einem kleinen Metallgehäuse, das am Druckerport angeschlossen wird. Dies dürfte so ziemlich die billigste Möglichkeit sein, einen voll Timecode-fähigen Sequencer zu bekommen.

Steinberg GmbH
Billwerder Neuer Deich 229
2000 Hamburg
Tel.: 0 40 / 769 25 16

Neues von C-Lab

Bei C-Lab, direkt gegenüber dem Steinberg Stand, gab es eine neue und stark verbesserte Version des Sequencers **'Creator'** zu sehen. Insbesondere der Event-Editor profitiert von den Verbesserungen. Er wurde jetzt mit einer Balkendarstellung versehen, die das Editieren sehr einfach macht. Ein Sequencer der Spitzenklasse.

Die Synthese-Software der amerikanischen Firma Digidesign, deren Übertragung auf den ST schon lange angekündigt war, ist jetzt lieferbar. Das Programm, das den Namen Softsynth trägt, wurde leider nicht gezeigt. Wir werden aber in Kürze näheres berichten.

Die ausgezeichneten Editoren von Digidesign werden, entgegen anderslautender Ankündigungen, nun wohl doch nicht auf den ST übertragen. Schade.

Auch der XAlyser, ein DX7-Editor mit ganz besonderen Features, wurde nicht gezeigt, geht aber jetzt wohl endgültig seiner Vollendung entgegen.

C-LAD Software
Friedrich-Ebert-Damm 204
2000 Hamburg 70
Tel.: 0 40 / 66 68 69

Eine spannende Ankündigung gab es bei MEV

Der Passport-Sequencer Master Tracks, den es schon seit einiger Zeit auf dem MacIntosh gibt, wurde auf den ST portiert. Es ist ein äußerst interessanter Sequencer, der eine extrem komfortable Benutzeroberfläche zu besitzen und dabei auch sehr leistungsfähig zu sein scheint. Das Programm besitzt 64 Spuren, für die es alle möglichen Editierfunktionen gibt. Besonders hübsch: Ein Controller-Editor, der die Controller-Daten wie eine Wellenform anzeigt. Eine gute Idee. Wir werden über das Programm berichten.

Außerdem gibt es bei MEV die Produkte von Dr. T's, über die wir zum Teil schon berichtet haben. Vom Notendruckprogramm 'The Copyist' gibt es eine neue Version, die aber gegenüber der alten Version nicht wesentlich verbessert wurde. Auch der Sequencer KCS wurde verbessert. Zu diesem Programm gibt es zwei Erweiterungen, die man als Kompositionshilfen bezeichnen kann. Die Eine, PVG für 'Programable Variations Generator' genannt, erzeugt aus vorhandenen Spuren Variationen, die Zweite die den Namen 'Fingers-The Algorithmic Composer' trägt, erzeugt aus interaktiv erstellten Algorhithmen neue Melodien. Leider konnten wir diese Programme noch nicht begutachten, auf der Messe war einfach keine Zeit dazu.

Schließlich gibt es bei MEV noch einen sehr hübsch gestalteten Editor für den DX711.

mev MidiSoft
Karl-Hromadnik-Str. 3
8000 München 60
Tel.: 0 89 / 83 50 31

Drei neue Editoren von PA Decoder

PA Decoder zeigte eine Serie aus drei Editoren für den Roland D-50, den Korg DS-8 sowie den Ensoniq ESQ-1. Der DS-8-Editor ist der einzige mir bekannte Editor für dieses Instrument.

Bei der Firma Ludwig, die auch eine Fertigversion des computergesteuerten Doepfer-Bausatzmischpultes vertreibt, konnte man einen Prototyp der ST-Software für dieses Gerät sehen. Auf dem Bildschirm werden die Fader dargestellt, Equalizer-Einstellungen werden über Kennlinien gezeigt. Das ist super! Wir hoffen, Ihnen auch dieses Gerät einmal in einem Test vorstellen zu können.

Ludwig Computer
Dingolstädter Str. 621
8000 München 45
Tel.: 0 89 / 311 30 66

Erstaunlich! – Es gibt ein neues GEM für den ATARI ST

Auf der ATARI Messe wurde von der holländischen Firma ABC-Software eine neue GEM-Version vorgestellt, die dem derzeitigen GEM auf den PCs entspricht. Die Firma, die sich mit GEM schon seit der ersten Stunde beschäftigt, hat von Digital Research, dem GEM-Entwickler, eine Lizenz für GEM erhalten. So wurde es dieser Firma möglich, exklusiv alle GEM-Applikationen an den ATARI ST anzupassen. Zuerst überlegte man, diese Programme unter dem ATARI ST-GEM zum Laufen zu bringen. Dies erwies sich als nicht praktikabel, da GEM erst richtig seine Möglichkeiten zeigt, wenn die Module vorhanden sind, die standardmäßig beim ATARI ST fehlen. Zum Beispiel erfordert die Installation zusätzlicher Gerätetreiber beim ATARI die Datei GDOS.PRG im Auto-Ordner. Programmen wie Easy Draw, Profi Paint usw. ermöglicht GDOS, Zeichensätze nachzuladen oder Grafiken auf beliebigen Ausgabegeräten, sprich Drucker, Plotter oder Datei auszugeben. Durch ein genormtes Dateiformat können Grafiken zwischen 'richtigen' GEM-Programmen ausgetauscht werden.

Unterschiede zum alten GEM

Als erstes fallen einem die äußeren Unterschiede auf. Die Fensterrahmen und Menüs sehen etwas anders aus. Dies alles sind Spätfolgen aus dem schon länger zurückliegenden Streit mit Apple. Diese Firma hatte damals einen Prozeß gegen Digital Research gewonnen, und Digital Research mußte als Folge sein GEM so verändern, daß die frappierende Ähnlichkeit mit dem Mac wegfiel. Dies hatte einige Nachteile zur Folge. So läßt sich zum Beispiel das Fensterformat nicht mehr stufenlos einstellen. Dafür wurden die GEM-Applikationen verbessert.

Im Folgenden mehr zu den internen Unterschieden:.

GEM System 2.2

Das neue GEM unterscheidet sich intern durch optimiertere Routinen. Unter anderem wurde das VDI überarbeitet. So wird der Grafikaufbau bei Benutzung dieses GEMs wesentlich schneller. Dies bedeutet aber nicht, daß Ihre alten GEM-Programme mit dem neuen GEM nicht mehr laufen. Im Gegenteil, einige Programme legen sogar geschwindigkeitsmäßig zu.

Das GEM System 2.2 wird auf dem ATARI ST beim Booten installiert. Es benötigt etwa 80 Kilobyte und setzt auf das alte Betriebssystem auf. So verwendet das VDI weiterhin Line A-Aufrufe und kann so auch den Blitter voll ausnutzen.

GEM Font und Drivers Pack

Zum neuen GEM-System sind diverse Druckertreiber verfügbar, die den jeweiligen Druckertyp und seine Grafikauflösung voll ausnutzen. Dies wird durch das unterstützte Metafileformat möglich. In diesen Files (Dateien) werden keine Pixelgrafiken, sondern Grafikbefehle gespeichert. Dadurch kann der jeweilige Druckertreiber alle Möglichkeiten des Ausgabegerätes nutzen, ohne daß die Quelldateien speziell für diesen Drucker erzeugt wurden. Bisher verfügbar sind Treiber für die Epson und NEC-Drucker. Besonders interessant ist es, daß bereits ein Treiber für den brandneuen ATARI Laserdrucker SLM 804 verfügbar ist.

GEM Desktop 2.0

Angepaßt an das neue GEM gibt es auch ein neues Desktop. Es präsentiert sich mit noch mehr Piktogrammen für die verschiedenen Applikationen (Programme unter GEM) und einen wesentlichen Vorteil: Alle Arbeitsabläufe lassen sich auch von der Tastatur steuern, und die Anmeldung von zusätzlichen Laufwerken entfällt. Die logischen Laufwerke bilden nun die oberste Ebene in der Dateihierarchie und werden automatisch angezeigt.

Endlich alle GEM-Software auch auf dem ATARI ST

Mit dem neuen GEM stellt ABC-Software nun auch alle GEM Programme von Digital Research vor. Es handelt sich dabei um Anwendungsprogramme, die alle miteinander zusammenarbeiten können. Hier eine Kurzbeschreibung der Programme, die bereits erhältlich sind:

GEM Graph

GEM Graph ist das Businessgrafikprogramm aus der GEM-Familie. Aus tristen Zahlen lassen sich blitzschnell und einfach, anspruchsvolle Grafiken erzeugen. Dabei kann man unter verschiedenen Kuchengrafiken, Histogrammen usw. in zwei- und auch in dreidimensionalem Format wählen. Um





die Übersichtlichkeit zu steigern, lassen sich verschiedene Darstellungsformen in einer Grafik kombinieren.

GEM Wordchart

Für die schnelle Erstellung von präsentativen Tabellen, Verzeichnissen und kleineren Publikationen bietet sich GEM Wordchart an.

GEM Draw Plus

Einigen ATARI Besitzern wird GEM Draw bereits bekannt sein. Obwohl dieses Zeichenprogramm nie im Handel war, befanden sich einige im Besitz dieses Programmes. Draw Plus dient der Gestaltung von objektorientierten Grafiken.

GEM Draw Plus Business Library

Im von GEM Draw verarbeitbaren Metafileformat sind auch schon vorbereitete Grafiken verfügbar. Unter dem Namen Draw Plus Business Library präsentiert sich eine Sammlung liebevoll gestalteter Grafiken, die man sofort in GEM-Draw Grafiken einbinden und bearbeiten kann. Das Metafileformat ermöglicht fast schon selbstverständlich auch die Verwendung in allen anderen GEM Applikationen.

GEM Write

Unter dem Namen GEM Write stellt sich eine Textverarbeitung vor, die mit verschiedenen Schriftarten arbeiten kann und es zuläßt, Grafiken aus GEM Draw oder GEM Paint einzubinden.

GEM Diary

Zusammen mit dem GEM Desktop und dem System wird GEM Diary geliefert. GEM Diary ist ein Deskaccessory zur Terminverwaltung. Enthalten ist eine Alarmuhr, die an Termine erinnert und ein fortwährender Kalender. Auf Karteikarten kann man sich Notizen machen.

GEM Fonteditor

Mit diesem komfortablen Malprogramm kann der Benutzer Zeichensätze editieren und im GEM-Format abspeichern. Diese Zeichensätze können dann von allen anderen Programmen verarbeitet und selbstverständlich auch gedruckt werden.

GEM Toolkit

In Kürze wird laut Ankündigung der Firma auch ein GEM Toolkit erhältlich sein. Dieses Toolkit ist für den Programmierer bestimmt, der Programme für GEM entwickeln will. In diesem Toolkit wird auch das Resource Construction Set II enthalten sein, das in Amerika bereits seit geraumer Zeit auch für den ST erhältlich ist.

Zukünftige Erweiterungen

Wie ein Gespräch mit dem Geschäftsführer der Firma ABC ergab, will man es nicht bei einer bloßen Anpassung des neuen GEMs an den ATARI belassen. Im Gegenteil: die Firma plant die Entwicklung spezieller Software für den ATARI ST. Unter anderem denkt man auch schon ernsthaft über Erweiterungen des GEM nach. Einige „Begrenzungen" ergeben auf einer so leistungsfähigen Maschine, wie dem ATARI ST, keinen Sinn mehr. So überlegt man bereits, die Anzahl der möglichen Fenster auf 12 oder 16 heraufzusetzen. Bereits in dieser Version ist es möglich, bis zu 12 Deskaccessories einzuladen.

Dokumentation, Vertrieb und Preis:

Falls Sie nun Appetit auf diese Software bekommen haben, hier sind die Preise:

GEM Draw Plus DM 359,–
GEM Draw Plus Business Library DM 169,–
GEM Wordchart DM 359,–
GEM Diary + GEM Desktop + GEM System DM 169,–
GEM Graph DM 359,–
GEM Fonteditor DM 359,–
GEM Write DM 399,–
GEM Font & Drivers Pack DM 169,–

Die Preise entsprechen leider nicht ganz dem, was man sonst von ST-Softwarepreisen gewohnt ist. Dafür erhält man eine wirklich ausgereifte Software mit einer deutschen Anleitung. Die Verhandlungen mit Vertriebspartnern in der Bundesrepublik waren zum Redaktionsschluß noch nicht abgeschlossen.

ABC Software
Dorpsstraat 2
NL-5314 AG Bruchem
Niederlande
Tel.: 00 31 / 41 04 / 0 11 (aus der BRD)

... und zum Schluß noch ein wenig Statistik

„Ihre Meinung interessiert uns", so lautete die Überschrift zu einem Fragebogen, der am Eingang der Halle 1 für die Besucher auslag. Kurz vor Redaktionsschluß erreichte uns das Ergebnis der knapp 1800 ausgewerteten Fragebogen. Dafür möchten wir uns bei der Werbeagentur Frenzel, Hopf und Partner GmbH recht herzlich bedanken. Im Folgenden nun die Antworten der ersten drei Fragen:

1. Wodurch sind Sie auf die ATARI-Messe aufmerksam geworden?

Tageszeitung:	17,38 %
Fachzeitschriften:	38,27 %
Plakate:	12,58 %
Händlerinfo:	7,03 %
Freunde/Bekannte:	15,04 %
Sonstige:	0,67 %

2a. Besitzen Sie einen Computer?

Ja:	89,03 %
Nein:	10,75 %

2b. Falls ja – welche Marke/Modell?

ATARI 400/600:	1,89 %
ATARI 800/130:	16,77 %
ATARI ST:	57,00 %
Sonstige:	14,87 %

3. Welche Anwendungsgebiete interessieren Sie auf der ATARI-Messe?

Textverarbeitung:	36,99 %
Datenbanken/Archiv:	22,79 %
Kfm. Abwicklung:	13,09 %
Techn. wiss. Anwendung:	27,24 %
Lernprogramm:	13,24 %
Unterhaltung/Hobby:	31,20 %

Angaben zur Person:

weiblich:	11,31 %
männlich:	87,08 %

Altersangabe (Jahren):

unter 14	5,29 %
15–17	10,14 %
18–24	26,24 %
25–35	29,53 %
über 36	27,63 %

Wenn man bei diesen Zahlen von einem repräsentativen Durchschnitt ausgeht, was bedingt durch die Anzahl der Fragebögen nicht sicher ist, ergeben sich teilweise recht interessante Ergebnisse. So sind z. B. die meisten Besucher ATARI-ST Besitzer, was nicht weiter verwunderlich ist. Aber, daß die am meisten vertretene Altersgruppe zwischen 25 und 35 Jahre alt ist, knapp gefolgt von denjenigen mit über 36 Jahren, zeigt doch wohl deutlich, daß ATARI sein Image als Spielecomputer Hersteller endgültig verloren hat. Bemerkenswert ist auch, daß Textverarbeitung mit 37 Prozent das mit Abstand häufigste Anwendungsgebiet ist.

(JW, CS, UB)

Last Not Least ... auch Spiele waren vertreten!

Spiele betreffend, war die Messe in Düsseldorf so üppig wie die Kollekte am Monatsende – gerade zweieinhalb renomierte Softwarehäuser waren mit Neuheiten vertreten.

Eigene Stände haben sich nur Electronic Arts und Microdeal (zusammen mit Michtron) geleistet. Rainbow Arts präsentierte die Atari ST Version von Bad-Cat – durch eine Verlosung von 20 Originalen – an unserem Stand am Sonntag nachmittag.

Besuch an unserem Stand hatten wir auch von Bomico. Die Firma von Adi Bolko bringt – rechtzeitig zum Weihnachtsgeschäft – Umsetzung der Comic-Abenteuer von Lucky Luck, Blueberry und Asterix.

Die ersten Spiele sollen bereits Ende Oktober erhältlich sein. Inhalt und Graphiken konnten wir bereits vorab in Augenschein nehmen.

Lucky Luke – Nitroglycerin:

Lucky soll den Bau der Transkontinentalen Eisenbahn beschützen, in einer Mischung aus Aktion und Strategie muß sich der Spieler mit schrägen Vögeln, Revolverhelden und den Brüdern Dalton auseinandersetzen.

Asterix im Morgenland:

Asterix und seine Freunde schweben per Teppich über die antike Welt von Griechenland, Persien und Rom. 1001 Stunden haben sie Zeit, um der schönen Prinzessin Rahazade zu helfen. Mit von der Segelpartie sind auch ihre alten Freunde – die Piraten „Una Salus Victis Nullam Sperare Saluten ..." (Vergll Enels)

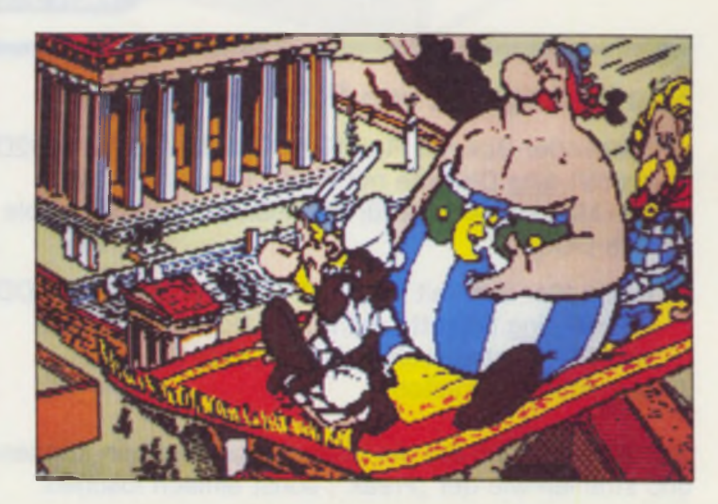
Blueberty, das Gespenst mit den goldenen Kugeln:

Im Gegensatz zu Lucky Luke und Asterix wurde hier nicht das neueste Comic-Heft umgesetzt. Die Geschichte ist Blueberty-Fans aus Heft 21 bekannt. Wer das Heft besitzt, findet darin nützliche Hinweise. Szenario ist ein fast verlassener Indianertreff im Grand Canyon. Als Regisseur und Held des Abenteuers zugleich versucht der Spieler das Gespenst mit den goldenen Kugeln zu besiegen.

(Preis je 64,95 DM: Anbieter Bomico, Elbingerstr. 3, 6 Ffm 90)

Altbekanntes im neuen Gewand gab es bei Electronic Arts zu bestaunen. Die Atari ST Versionen von The Bard's Tale und Marble Madness sind endlich da.

Das Rollenspiel, in dem der Barde und seine Freunde die Stadt Skara Brae von bösen Zauberer Manger befreien, ist sehr gut gelungen. Eher enttäuschend, besonders graphisch, war dagegen das Wiedersehen mit der verrückten Murmel. Rechnet man die lange Wartezeit dazu, wird Marble Madness dem hohen Standard, den ECA-Produkte normalerweise haben, nicht gerecht. Fragern nach weiteren Neuerscheinungen aus dem Hause Electronic Arts wurde ein freundlicher Hinweis auf die in London stattfindende PCW gegeben. Mehr darüber also in unserem nächsten Heft.





SALIX PROLOG FÜR ALLE ATARI ST

mit ausführlichem deutschem Handbuch

LEISTUNGSDATEN:

- ⋆ **Compiler/Interpreter System.** Der inkrementelle Compiler erzeugt einen Zwischencode, der vom Interpreter abgearbeitet wird. Geschwindigkeit 1200 Lips.
- ⋆ **Edinburgh Standard.** Alle im Buch 'PROLOG – Programming in Logic' von Clocksin & Mellish definierten Funktionen sind verfügbar.
- ⋆ **Zusätzliche eingebaute Funktionen,** (primitives), insgesamt ca. 140 Funktionen
- ⋆ **Integer- und Gleitkommaarithmetik,** mathematische Funktionen, Gleitkomma in doppelter Genauigkeit.
- ⋆ **Syntax Error Corrector.** Tritt beim Einlesen eines PROLOG-Ausdrucks (vom Bildschirm oder vom Laufwerk) ein Syntaxfehler auf, erhält der Benutzer Gelegenheit, den Fehler am Bildschirm zu verbessern.
- ⋆ **Lector.** Ein in PROLOG geschriebenes Hilfsmittel, um PROLOG-Programme auf Tippfehler und falsche Verwendung der eingebauten Funktionen zu prüfen.
- ⋆ **Anschluß an einen vorhanden Editor.** (Z. B. 1st-Word) Die PROLOG-Sitzung muß nicht verlassen werden, um eine Bibliothek zu editieren.
- ⋆ **GEM-Library** gestattet die Nutzung fast aller GEM-Funktionen von PROLOG aus, damit steht die riesige Grafikbibliothek des GEM in der Form von PROLOG-Prädikaten zur Verfügung.
  Zusätzlich Window Management System mit Pufferung und automatischem Refresh der Fenstertextinhalte.
- ⋆ **GEM-Top-Level-Interpreter.** Komfortable Bedienung des PROLOG-Programmiersystems mit Drop-Down-Menüs und Formularen. Der Gem-Top Level Interpreter ist komplett in PROLOG geschrieben, damit auch für spezielle Anforderungen modifizierbar.
- ⋆ **Startup-File** um das PROLOG-System entsprechend den jeweiligen Anforderungen zu konfigurieren.
- ⋆ Der **volle Adressraum des 68000** (auch 4 MB) wird unterstützt, so volle Ausnutzung des ATARI-Speichers und etwaiger Speichererweiterungen.
- ⋆ **Deutsches oder englisches Handbuch** mit mehr als 300 Seiten

Handbuch einzeln 60,– DM wird beim Kauf angerechnet

Preise sind unverbindlich empfohlene Verkaufspreise



DIE NEUE VERSION VON SALIX PROLOG IST DA!

- ⋆ SALIX PROLOG 2 macht das Programmieren in PROLOG noch einfacher und schneller. Dafür sorgt schon der eingebaute Datenbankeditor. Dieser Editor erlaubt es, die im Speicher befindlichen Regeln auf einfache Weise zu modifizieren. Dabei wird gleich die Syntax geprüft. Eine Fülle von Kommandos – sowohl über Tastatur als auch über Drop-down-Menüs abrufbar – erleichtert die Navigation durch die Datenbank. Und hat man sich einmal vertippt, macht die UNDO-Funktion die letzte Operation rückgängig.
- ⋆ Tritt während des Ablaufs eines PROLOG-Programms ein Fehler auf, so wird die fehlerhafte Regel in den Editor geholt und kann sofort korrigiert werden.
  Hat man sich bei einer Abfrage einmal vertippt, so bringt die HELP-Taste die letzte Eingabe zur Abänderung noch einmal auf den Schirm.
- ⋆ Die Programmentwicklung wird durch erweiterte Debug-Möglichkeiten unterstützt. So gibt es jetzt die Möglichkeit, den Programmablauf während des Testens abzuändern, von einem bestimmten Punkt aus den Trace noch einmal zu wiederholen, oder alle Traceinformation nicht auf dem Bildschirm auszugeben, sondern in eine Datei zu schreiben.
- ⋆ Mit zusätzlichen eingebauten Funktionen eröffnet SALIX PROLOG 2 ganz neue Möglichkeiten für PROLOG Programme. Insgesamt sind es jetzt über 150 eingebaute Funktionen. Insbesondere die block- und blockexit-Funktionen und die Verarbeitung unendlicher Datenstrukturen (und das in endlicher Zeit!) sollen hier genannt werden. Diese Funktionen stammen aus dem Marseille-Prolog und waren bisher im Edinburgh-Standard nicht vorhanden.
- ⋆ Auch für das Arbeiten mit GEM sind Erweiterungen vorhanden: So können dynamisch Formulare aufgebaut werden, benutzerdefinierte Kommandos können in die Pull-down-Menüs eingebaut werden. Die Verarbeitung von Bild-Dateien gestattet es, Expertensysteme mit erläuterten Bildern zu versehen.
- ⋆ **SALIX PROLOG 2 läuft auf allen Rechnern der ATARI-ST-Serie mit jedem Speicherplatzausbau.**

UP-DATE-SERVICE

Registrierte Benutzer von SALIX-PROLOG 1 erhalten SALIX PROLOG 2 zum Sonderpreis von 99,– DM (Registrierkarte und Originaldiskette zurücksenden)

Heim-Verlag
Heidelberger Landstr. 194
6100 Darmstadt-Eberstadt
Telefon 0 61 51 / 5 60 57

BESTELL-COUPON einsenden an: **Heim**-Verlag
Heidelberger Landstr. 194 · 61 Darmstadt-Eberstadt

Bitte senden Sie mir:

_____St. **SALIX-PROLOG** mit über 300 S. deutschem Handbuch

_____St. **SALIX PROLOG** mit über 300 S. englischem Handbuch

zum Preis von 198,– DM je Stück
zzgl. DM 5,– Versandkosten bei NN

☐ per Nachnahme ☐ Verrechnungsscheck liegt bei

Name: ______________________

Straße: ______________________





Mit der gleichen Begründung hat auch Microdeal in Düsseldorf nicht alle Karten auf den Tisch gelegt. John Symes neueste Produkte in Düsseldorf waren das Airball Construction Kit und Tanglewood Cader „was macht der Roboter ohne Kleingeld in der Telefonzelle?" Mit dem Hinweis auf eine Präsentation bei der PCW wurden noch die Spiele Fright Night, ST Soccer, Omega Run, Fatherneck genannt. Am späten Sonntag nachmittag – der Besucherstrom war bereits von der angebrochenen Teestunde gekennzeichnet – gab es dann noch ein Bonbon für Zurückgebliebene: Goldrunner 2 -- Strategie statt Artillerie ist diesmal die Devise. Den Weg zum Treibstoff für das Fluomobil muß man sich erst in einem komplizierten Labyrinthspiel bahnen.

(CPL)

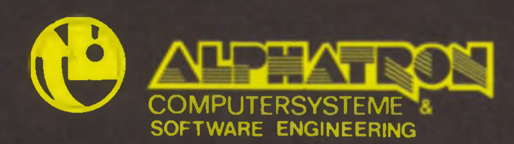


DM 99,–

für

ATARI ST

- **HAMLET II** ist schneller durch einen neuen, selektiven Algorithmus
- **HAMLET II** beherrscht alle Remis-Regeln und Unterverwandlungen
- **HAMLET II** ist voll GEM + Maus gesteuert
- **HAMLET II** speichert komplette Partieen mit Kommentartext in einer Bibliothek
- Leckerbissen: — beliebige Spielstufenwahl durch Zeitvorgabe
  — integrierte Eröffnungsbibliothek
  — der Rechenvorgang kann auf dem Brett dargestellt werden



LUITPOLDSTRASSE 22 · 8520 ERLANGEN · RUF 09131/25018

The MOUSE-PAD™

- ☐ Die praktische und rutschfeste Tischauflage für die Computer-Mouse.
- ☐ Exaktes *„Fahren"* ist nun auch auf unebenem Untergrund möglich.
- ☐ Erhöht die Lebenszeit der Mouse und schont die Mechanik.
- ☐ Lieferbar in den Farben Mittelblau oder Pink, Format: 270 × 220 mm.

Nur DM 19,80 empf. Verkaufspreis
Händleranfragen erwünscht.

MSM GmbH · Bayenthalgürtel 45 · 5000 Köln 51 · Tel. 02 21/38 20 20

COMPUTERVERSAND WITTICH

Tulpenstr. 16 · 8423 Abensberg

☎ **0 94 43/453**



Atari 520 STM	498,–
Atari 1040 STF	998,–
Mega ST	auf Anfrage
Atari SH 205	1198,–
Aufrüstung auf 1 MB	198,–
Monitor SM 124	398,–
Original Maus	98,–
Disk. Station SF 354	169,–
NEC Disk. Stat. 720 KB	369,–
Scart Kabel	38,–
Dataphon S21/230	398,–
1st Teacher	49,–
Farbmonitore	ab 498,–
Drucker	
STAR NL 10	598,–
NEC P6	1111,–
NEC P7	1498,–

Public-Domain Softwarepakete

★ **5 Markendisketten MF 1 DD, doppelseitig formatiert und gefüllt mit guter Public-Domain Software**



Paketpreis nur DM 45,–
inklusive Porto und Verpackung!

★ Weihnachtsangebot ★

3 Pakete nur DM 119,–
6 Pakete nur DM 219,–
inklusive Porto und Verpackung

- ☐ 1 PD01 – PD10 aus ST-Computer
- ☐ 2 PD11 – PD20 aus ST-Computer
- ☐ 3 PD21 – PD30 aus ST-Computer
- ☐ 4 10 PD-Disk aus Kalifornien
- ☐ 5 10 PD-Disk aus Kalifornien
- ☐ 6 10 PD-Disk aus aller Welt
- ☐ 7 10 PD-Disk aus Kalifornien
- ☐ 8 10 PD-Disk aus Kalifornien
- ☐ 9 10 PD-Disk aus Kalifornien
- ☐ 10 PD31 - PD40 aus ST-Computer
- ☐ 11 PD41 - PD51 aus ST-Computer
- ☐ 12 10 PD-Disk aus ST-Computer
- ☐ 13 Paket Enthält bewegte Comics ★ Formatierprogramm bis 85 Tracks ★ Disk Magazin ★ Accessories ★ **Schachprogramm** etc...
- ☐ 14 Paket Shuttle Bilder ★ Bildverarb. Demo ★ Sample 3 Musik ★ dazu **die neuen PD Programme aus Kalifornien** ★ Swop-Shop, intern. Anzeigenbörse, u.v.m...
- ☐ 15 Paket PMU verwaltet Ihre Print Master picture libraries, dazu gibt's noch 189 neue Bilder ★ neue Scenarios für FlightSim II ★ Textadventures und Strategiespiele, natürlich mit Source ★ Archive und TinyStuf, beide in der neuen Gem-Version ★ u.v.m.
- ☐ 16 Paket Enthält PD 52 – 61 aus ST-Computer. Fugger ★ Ausland ★ Kepler ★ Eti-Master ★ Trio ★ Kerne ★ Meßwert. etc...
- ☐ 17 Paket Enthält PD 62 – 71 aus ST-Computer. E-Plan ★ Paintlux ★ Degas Elite Fontmaker ★ Vang Gogh ★ Orbit ★ Roulette u.v.m.
- ☐ 18 Paket ★ Label, schönes Etikettprg. ★ GfA ★ Logo Prgs ★ Meteosat Diashow ★ Kissdemo ★ Heinzelmann ★ Trek 2 Textadventure ★ Miami Digisound (1MB) ★
- ☐ 19 Paket Enthält PD 72 – 81 aus ST-Computer. Z. B. Videoarchiv ★ Maxidisk ★ Datebook ★ Periode ★ AESLIB ★ U_Boot ★ Fonts für PD 40 ★ Malprogramm ★ etc...
- ☐ 20 Paket Enthält PD 82 – 91 ★ Updates „U1" und „U2" aus ST-Computer, z. B. Skat, Grusel, Masterpainter, Starofix, Hypervoc, etc...

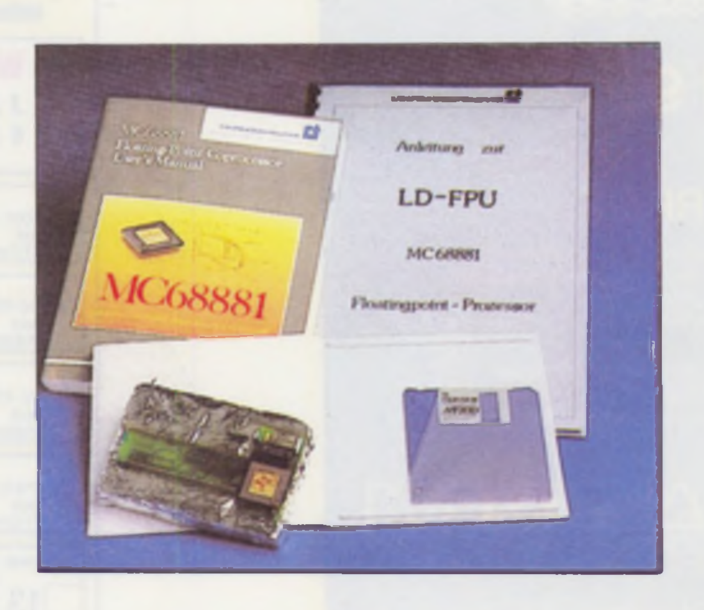
Gewünschte Pakete ankreuzen u. Bestellung einsenden an:

IDL-Software
– Public Domain –
Alkmaarstraße 3 · 6100 Darmstadt 13

- ☐ Info-Diskette Nr. 3 DM 5,– PD-Liste, 180 neue Levels für Arkanoid + Utility zum Erstellen eigener Levels
- ☐ SCHECK über DM .... liegt bei. **Ich erhalte die Pakete verpackungs- u. versandkostenfrei**
- ☐ Nachnahme (Nur Inland, zuzügl. DM 5,– NN-Gebühr)

Der ATARI ST bekommt Nachhilfe in Mathematik

1. Teil: Die Hardware



Seit einiger Zeit taucht verstärkt der Name MC 68881 in der Presse auf, womit der Floating Point Coprozessor der 68000-Familie von Motorola gemeint ist. Seit nunmehr einigen Wochen wird diese Erweiterung auch für den ST angeboten. Was ein Coprozessor ist, ob und wie man ihn einsetzen kann, und was er für Vorteile bringt, soll in dieser und der nächsten Ausgabe genauer erläutert werden.

In dieser Ausgabe beschäftigen wir uns vorwiegend mit der Hardware, während wir nächsten Monat auf die unterstützende Software eingehen, dabei werden auch die Programmiersprachen unter die Lupe genommen, die die FPU (***F**loating **P**oint **U**nit*) schon unterstützen. Auch die berühmten Benchmarks werden nicht fehlen.

Alle Mikrocomputersysteme sind nach einem ähnlichen Schema aufgebaut, wie es in Bild 1 zu finden ist. Dieser Mikroprozessor hat einen Taktgeber, eine Befehlsdekodierung, ein internes RAM, womit die Register des Prozessors gemeint sind, ein internes ROM, in dem das Mikroprogramm (Firmware) vorhanden ist, und eine ALU – eine **Arithmetisch Logische Einheit** (Unit). Dieser Teil des Prozessors ist dem Durchführen von arithmetischen und logischen Verknüpfungen gewidmet, d. h. er hat die Aufgabe – bei einfacheren Prozessoren – eine Addition oder Subtraktion oder – bei komplexeren Prozessoren wie dem 68000 – sogar Multiplikationen, beziehungsweise Divisionen durchzuführen. Obwohl diese Unterstützung des Prozessors im Vergleich zu älteren Prozessortypen, wie zum Beispiel dem 6502 in Bezug auf die Multiplikation schon enorm ist, darf nicht vergessen werden, daß dies nur für den Bereich der GANZEN Zahlen zutrifft!

Diese beachtenswerte Einschränkung findet ganz besonders im wissenschaftlichen Bereich (Universitäten) der Rechnernutzung wenig Zuspruch. Dort ist man ganz besonders auf schnelle Berechnungen angewiesen, wobei Kommazahlen eine überragende Bedeutung haben – man denke hier zum Beispiel an die Fourier-Analyse oder zum Beispiel an CAD, das Computer Aided Design. Die Wichtigkeit der Kommazahlen zeigt sich besonders in der Geschwindigkeitsangabe der Großrechner, die nicht mehr in normalen OPs (operations), sondern in sogenannten FLOPs angegeben werden, den ***Fl**oating Point **OP**erations*, also Fließkomma-Operationen.

Es leuchtet sicherlich ein, daß die Verarbeitung von Kommazahlen ein größeres Maß an Mehraufwand erfordert, als die einfche Behandlung von ganzen Zahlen. Was also liegt näher, als der eigentlichen ALU des Prozessors eine Hilfe zukommen zu lassen. Dies wird durch einen sogenannten Coprozessor erreicht. Bei der INTEL-Familie der 8088, 8086 etc. findet man beispielsweise den 8087, während MOTOROLA die 68000-Familie mit ihrem MC 68881 bereichert.

Was ist ein Coprozessor?

Will man eine hohe Effizienz erreichen, so muß man versuchen, Prozessor und Coprozessor möglichst direkt miteinander kommunizieren zu lassen. Motorola hat deshalb sein Flagschiff, den MC68020, also die 32Bit-Version des MC68000, optimal auf den MC68881 abgestimmt. Die Register der FPU können direkt im MC 68020 angesprochen werden, obwohl

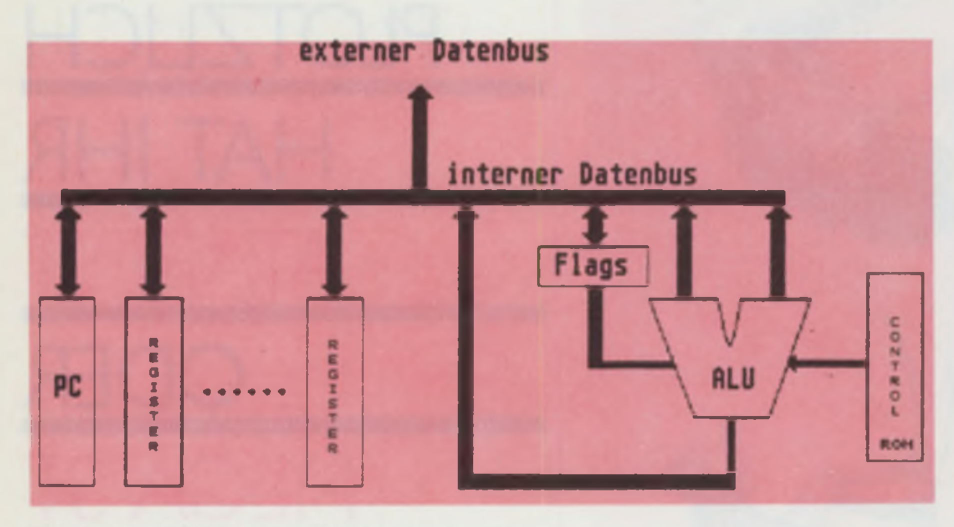


Bild 1: Aufbau des Mikroprozessors

sie sich in der Realität im Coprozessor befinden. Das bedeutet, daß sich die Zusammenarbeit von Prozessor und Coprozessor für den Programmierer dahingehend äußert, daß er das Gefühl hat, Prozessor und Coprozessor seien eine Einheit. Dabei erfolgt der Zugriff über ein bestimmtes Busprotokoll, das über den Line-F-Trap geregelt wird, den der MC68020 bezogen auf den MC68881 hardwaremäßig unterstützt. Diese Unterstützung fehlt bei den „kleineren" Typen der 68000-Familie, so daß ein Floating Point Befehl einen TRAP-Sprung auslöst, der dann die Argumente des Befehls an den Coprozessor weiterleitet und damit den Zugriff auf den Coprozessor regelt.

Würde man also solange wie kein Coprozessor vorhanden ist, Software-Routinen zur Berechnung von Fließkommaoperation im Betriebssystem über Line-F einbinden, könnte man mit der Änderung eines Vektors bei vorhandenem Coprozessor allen(!) Programmen die Schnelligkeit des Coprozessors zukommen lassen. So ist es beim ATARI nicht geschehen: Leider wurde der Line-F-Trap nicht für das MC68881-Coprocessor-Interfacing vorgesehen, sondern für Grafikzwecke eingesetzt – so werden zum Beispiel AES-Routinen über Line-F aufgerufen. Dennoch kann die FPU verwendet werden, da sie auch als Portbaustein angesprochen werden kann und der Line-F-Trap nicht zwingend ist.

Also doch kein Coprozessor im ST!

Obwohl durch diesen schon benutzten Line-F-Trap einiges an Eleganz verloren geht, bedeutet es nicht, daß die im ST verwendete Methode langsamer ist! Vielmehr werden die Adressen direkt angesprochen, dies wird an einem späteren Beispiel näher erläutert. Diese Tatsache bedeutet nicht, daß die FPU dadurch nicht mehr als Coprozessor wirkt, sie wird aber nicht wie üblich angesprochen. Da der MC-68000 nicht wie der 68020 selbständig entscheiden kann, ob die Fließkommaoperation durch Software oder durch die FPU verarbeitet werden soll, ist die erreichte Geschwindigkeitssteigerung gleich NULL, so daß vorhandene Software ohne Anpassung um keine einzige Millisekunde schneller werden. Einige Firmen sind allerdings schon dabei, ihre Programme speziell auf die FPU anzupassen, so auch einige Programmiersprachenhersteller.

Bedenkt man, daß trigonometrische Funktionen wie beispielsweise SIN und COS durch eine Reihenentwicklung (Taylor) erzeugt werden, so kann die Leistung eines Floating-Point-Coprozessors nicht unterschätzt werden. Programmiert man diese selbst, sind Zeiten von 10 Millisekunden keine Seltenheit. Nicht zu vergessen ist die sehr hohe Genauigkeit (80 Bit-Fließkomma!, das DOUBLE-Format in diversen Compilern hat 64 Bit), die der MC68881 besitzt und die trotz allem kaum die Geschwindigkeit beeinflußt. So dauert ein Sinus auf dem ST mit FPU, die mit 16 MHz, man beachte, doppelt so hoch getaktet ist wie die CPU, circa 29 Mikrosekunden, was einer Steigerung von etwa Faktor 350(!) auf die oben erwähnten 10 Millisekunden entspräche (Dieses Beispiel ist sogar noch untertrieben, wie Sie nächsten Monat anhand der BENCHMARK-Test erkennen werden). Zieht man nicht zuletzt das vereinfachte Block-Diagramm (Bild 3) der FPU in die Betrachtungen ein, so ist es sicherlich nicht schwer zu akzeptieren, daß die FPU gar kein komplexerer Chip als die eigentliche CPU MC68000 ist!

Die LD-FPU

Die Firma LISCHKA DATENTECHNIK ist der erste Anbieter einer Fließkomma-Einheit, die direkt für den ST erstellt wurde. Deren Coprozessorplatine liegt uns als Exemplar vor. Der Lieferumfang des Paketes ist, was nicht unbedingt der Regel entspricht, als äußerst zufriedenstellend zu bezeichnen. Neben der eigentlichen Platine und natürlich dem Coprozessor selbst, liefert Lischka Datentechnik „mehr als ein Pfund Dokumentation" (Zitat aus der Produktbeschreibung der Firma). Dieses Pfund besteht aus einer aus-

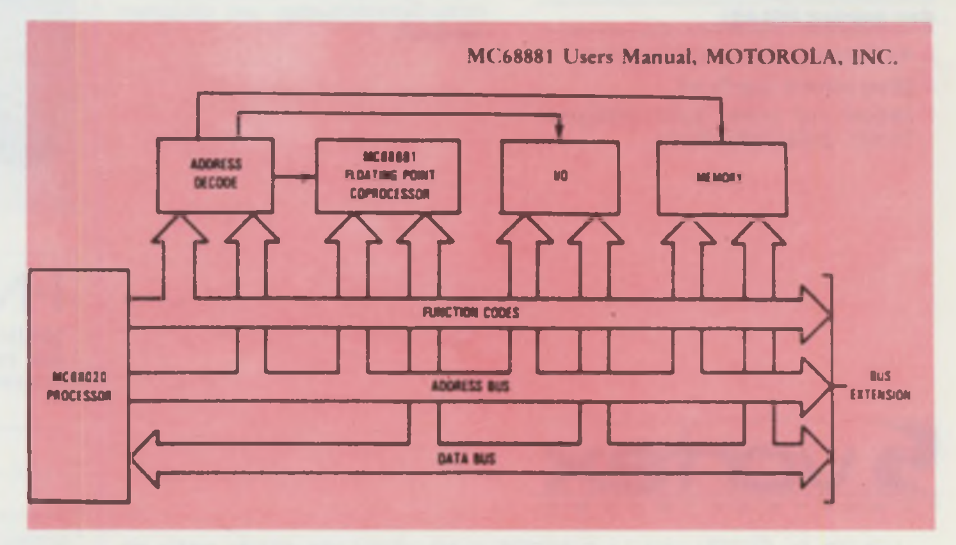


Bild 2: Typische Coprozessor Konfiguration

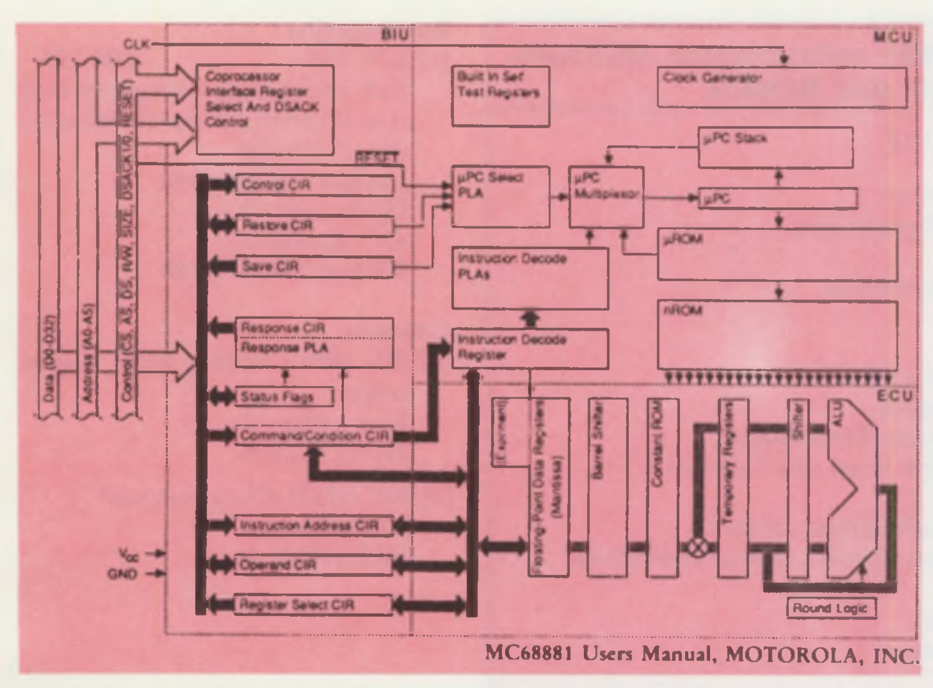


Bild 3: Der vereinfachte Aufbau der FPU MC 68881

führlichen Einbaubeschreibung, sowie einer Einführung in die Programmierung des Floating Point Coprozessors. Außerdem wird das sehr gute (und nicht billige) von MOTOROLA herausgegebene USER's MANUAL des MC68881 mitgeliefert, das auch die letzten Fragen des Programmierers und Entwicklers beantworten wird. Mehrere Libraries für Compilersprachen sind zur Nutzung des MC68881 auf einer Diskette vorhanden.

Der Einbau der LD-FPU

Der Einbau der LD-FPU ist in mehreren Varianten möglich, bei denen man aber schon ein wenig Geschick aufbringen sollte. Für diejenigen ST-Besitzer, deren Prozessor gesockelt ist, gestaltet sich der Einbau recht einfach, da sie nur den Prozessor heraushebeln, die Platine einsetzen und den Prozessor auf die Platine setzen müssen. Danach muß noch ein PIN der CPU bearbeitet werden (BERR), aber dazu unten mehr.

Ist der Prozessor gesockelt, so wird es ein wenig knifflig. Dann nämlich muß die Platine direkt auf den Prozessor gelötet werden, was handwerkliches Geschick voraussetzt, da die Lötstellen nicht zu lange erhitzt und auch keine Verbindungen zwischen den Pins hergestellt werden dürfen. Anstatt des direkten Auflötens der LD-FPU-Platine, kann auch zuerst ein Sockel auf die CPU gelötet und dann die Platine in den Sockel gesteckt werden. Der Nachteil dieser Methode ist, daß der Platz im Atari ST sehr eingeschränkt ist und beim Einbau des Abschirmblechs darauf geachtet werden muß, daß dieses den Koprozessor nicht berührt.

Auf eine Nachfrage hin, soll hier darauf hingewiesen werden. Der Steckplatz der LD-FPU sollte möglichst freigelassen werden, denn ATARI hat angekündigt, ihren Blitter im 68000er-Gehäuse auszuliefern. LD-FPU-Besitzer müßten demnach ihren Blitter dann nur noch in die Fassung einsetzen.

Außer dem Einbauen der eigentlichen LD-FPU-Platine muß noch ein weiterer Arbeitsgang durchgeführt werden. Dieser betrifft PIN 22 der CPU. Der PIN 22 entspricht dem BUS-Error-Signal und muß, aufgrund des neugenutzten Speicherbereiches, über die FPU-Platine und dann an die CPU geführt werden. Die BERR-Leitung gibt genau dann eine Meldung aus, wenn Zugriffe auf Speicherbereiche, die gesperrt oder nicht vorhanden sind, stattfinden. Die FPU befindet sich aber nicht im normal verfügbaren Adressraum, so daß der GLUE-Chip dies bemerkt und eine Fehlermeldung über die BERR-Leitung an die CPU weitergibt. Da diese Leitung bei Umgestaltung aber nun über die FPU-Platine führt, wird die Bus-Error-Meldung nur dann weitergegeben, wenn sie sich nicht auf die FPU bezieht.

Der Test der ganzen Einheit wird schließlich durch fertige, mitgelieferte Test- und Demoprogramme erleichtert, so daß der Käufer sicher sein kann, daß der Einbau erfolgreich verlaufen ist.

Der professionelle Einbau ist recht preiswert

Sicherlich mag dieser Arbeitsaufwand einigen Käufern zu hoch sein, aber man hat für Abhilfe gesorgt: LD bietet dem Kunden einen kompletten Einbau inclusive Versand für nur DM 69,– Aufpreis – eine preislich interessante Alternative.

Die Ablösung für den MC 68881 ist der MC 68882

Zum Schluß dieses Artikels möchte ich noch auf einen Leckerbissen hinweisen, den Motorola für das 1. Quartal 1988 angekündigt hat: den MC68882. Dieser Floating Point Processor wird PIN- und Funktionskompatibel zum MC68881 sein, so daß er einfach austauschbar ist! Er ist von Hause aus um einiges schneller als sein kleinerer Bruder, was unter anderem an der verbesserten und erweiterten internen Parallelverarbeitung liegt. Eine Nachfrage bei Lischka Datentechnik ergab, daß die neue FPU auch für den ST erhältlich sein wird sobald Stückzahl und Preis akzeptiert werden können.

Ganz aktuell: Ab sofort muß die BERR-Leitung nicht mehr bearbeitet werden, da ATARI CORP. die Leitung des GLUE als OPEN-COLLECTOR beschrieben hat. Dadurch entfällt ein weiterer Arbeitsgang beim Einbau der LD-FPU, der dadurch noch einfacher wird!

Eine weitere brandaktuelle Meldung ist, daß der **Blitter** tatsächlich im Gehäuse von der gleichen Größe des 68000 in Zusammenhang mit neuen ROMs ausgeliefert wird, wobei uns als Datum etwa Mitte November mitgeteilt wurde – LD-FPU-Besitzer müssen diesen dann nur auf die FPU-Platine aufstecken.

Übersicht 1: Die vom MC 68881 unterstützten arithmetischen Funktionen und Befehle

Befehl	Funktion
FABS	Absolutbetrag
FCOS	Cosinus
FACOS	Arcus Cosinus
FCOSH	Cosinus Hyperbolicus
FSIN	Sinus
FSINH	Sinus Hyperbolicus
FASIN Arcus Sinus	
FTAN	Tangens
FTANH	Tangens Hyperbolicus
FATAN	Arcus Tangens
FATANH	Arcus Tangens Hyperbolicus
FSINCOS	gleichzeitiger Sinus und Cosinus!
FTENTOX	10^x
FETOX,FETOXM1	Exponentialfunktion, minus 1
FLOGN,FLOG2,FLOG10	Natürlicher-, Zweier-, Zehnerlogarithmus
FADD	Addition
FSUB	Subtraktion
FMUL	Multiplikation
FDIV	Division
FSQRT	Quadratwurzel
FMOD	Modulo-Funktion
FNEG	Negieren
FTWOTOX	2^x
FSCALE	2^x mit Integer
FBcc, FDBcc	Bedingter Sprungbefehl, mit Dekrement
FTRABcc	Bedingter TRAP
FCMP	Fließkommazahl-Vergleich
FGETEXP	Ermitteln des Exponenten
FGETMAN	Ermitteln der Mantisse
FINT, FINTRZ	Ermitteln der Ganzzahl, gerundet

und andere...

Im FPU-ROM enthaltene Konstanten:

PI, $LOG_{10}(2)$, e, $LOG_2(e)$, $LOG_{10}(e)$, 0.0, LN(2), LN(10)
$10^0 10^1, 10^4 10^8 10^{16}, 10^{32}, 10^{64},$
$10^{128} 10^{256}, 10^{512}, 10^{1024}, 10^{2048}, 10^{4096}$

Bezugsquelle:

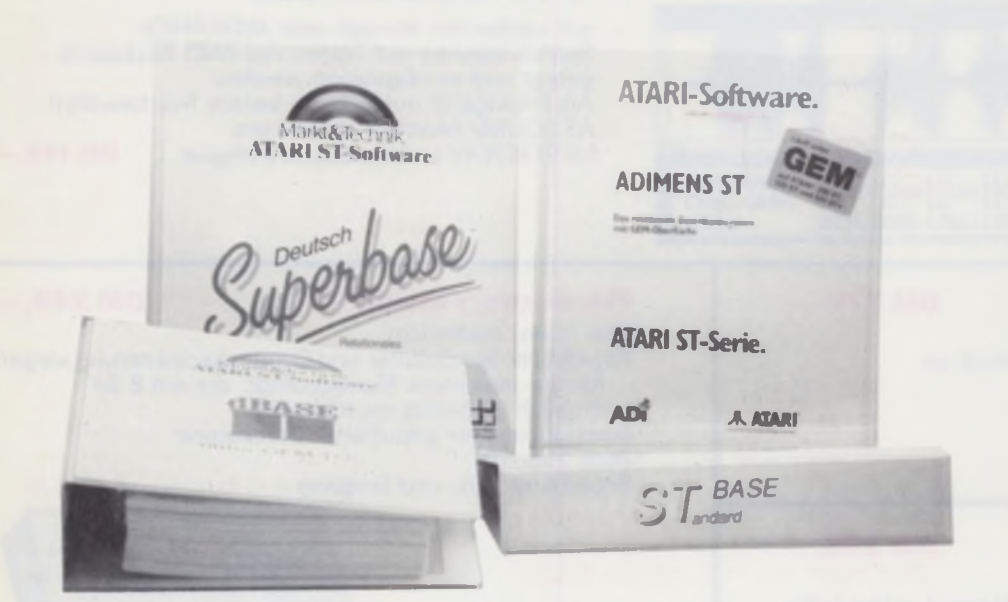
[1] LD-FPU DM 729,-
incl. reichhaltiger Dokumentation u. a. [2] und Libraries für verschiedene Compiler
Lischka Datentechnik
Hochstraße 22
4173 Kerken 2
Telefon 0 28 33 / 73 88

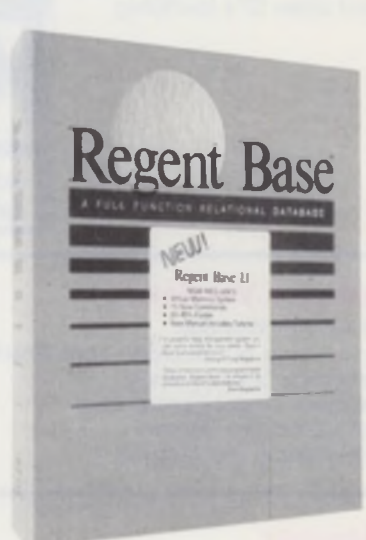
Literatur:

[2] MC 68881 Floating Point Coprocessor
User's Manual
MOTOROLA

[3] MC 68020 Microprozessor
User's Manual
MOTOROLA

Relationale Datenbanken unter sich
– Ein Anwender-Vergleichstest





Ein häufiges Anwendungsgebiet von Personal Computer ist die Sammlung großer Datenmengen. So ist es auch nicht verwunderlich, daß schon gut ein Dutzend Programme zu diesem Thema existieren. Wir testeten fünf relationale Datenbanken unter GEM, die alle mit deutschem Handbuch geliefert werden. Sie besitzen alle die Mindestfunktionen, die für einen professionellen Einsatz erforderlich sind.

Mit Relationen

Was unterscheidet relationale Datenbanken von gewöhnlichen Dateiverwaltungen und was sind die Vorteile eines solchen Systems? Nun soll zunächst geklärt werden, was eine relationale Datenbank ist. Dazu möchte ich Ihnen hier die drei gebräuchlichsten Datenbankstrukturen vorstellen. Ziel bei der Erstellung einer Datenbank ist, neben einer schnellen Zugriffsgeschwindigkeit auf die Daten, vor allem die Integrität der Daten und damit auch die Vermeidung von Redundanz. Integrität beeutet, daß innerhalb der Datenbank keine Widersprüche oder Fehlinformationen auftreten. Angenommen in einer Datei sind Rechnungsdatei und in einer Auftragsdatei sind Adressen gespeichert und Sie bekommen die Mitteilung, die Adresse von xyz hat sich geändert. Normalerweise müßte man hier nachschauen, wo überall xyz auftaucht und gegebenenfalls korrigieren. Es besteht die Gefahr, Daten zu übersehen (auch bei Computerunterstützung); außerdem enthalten die Dateien Redundanz, d. h. es hat ja eigentlich keinen Sinn eine Adresse mehrfach zu speichern.

Also geht man bei Datenbanksystemen meist den Weg, nur einen Verweis in die Adressdatei zu speichern. So genügt in diesem Fall die Änderung dort.

Das hierarchische Modell

Bei diesem Verfahren werden die Daten hierarchisch geordnet. Die Datenbank wird aus einem Baum gebildet. Die Wurzel ist die Datenbank selbst. Sie verzweigt dann zum Beispiel in die verschiedenen Rechnungen. Jeder Rechnungsdatensatz hat nun wieder Unterobjekte, wie zum Beispiel eine Adresse. Da die Beziehung innerhalb der Datenbank nur immer in eine Richtung möglich ist, ergeben sich einige Nachteile. Das Problem der Redundanzspeicherung und der Integrität läßt sich mit einem Trick lösen.

Kommt eine Adresse mehrfach vor, so wird sie nur einmal gespeichert. An al-

len anderen Stellen stehen virtuelle Datensätze, die intern als Verweis gespeichert werden.

Das Netzwerk Modell

Dieses Verfahren unterscheidet sich in der Art der Datenverknüpfung nicht vom hierarchischen Modell. Es besteht jeweils nur eine Verbindung zwischen zwei Datensätzen. Nur werden hier die Daten nicht in Baumform organisiert, sie sind eher willkürlich miteinander verbunden und bilden so ein Netz miteinander verbundener Datensätze.

und last but not(!!) least, das relationale Modell

Nun endlich zum relationalen Modell. Nachdem Sie bereits einiges über die anderen Verfahren wissen (und über deren Nachteile), nun zur Idee der relationalen Datenbank. Alle Daten werden hierbei in Tabellen untergebracht. Als Überschrift hat diese Tabelle die Merkmalsnamen. Für eine Adressdatei in etwa so:

Name, Vorname, Anrede,
Straße, PLZ, Wohnort

Darunter befinden sich dann zeilenweise die Datensätze, sprich die Informationen. Bisher geht so etwas auch mit Karteikarten. Interessant wird es erst, wenn mehrere Dateien gemeinsam eine Datenbank bilden. Eine zweite Datei könnte zum Beispiel alle offenen Rechnungen enthalten, und Verweise in die Adressdatei. Dieser Verweis kann eine Kundennummer, oder irgend ein anderes eindeutiges Merkmal sein. Dies ist dann der **'Schlüssel,** der die Verzweigung von der Rechnungsdatei in die Adressdatei ermöglicht.

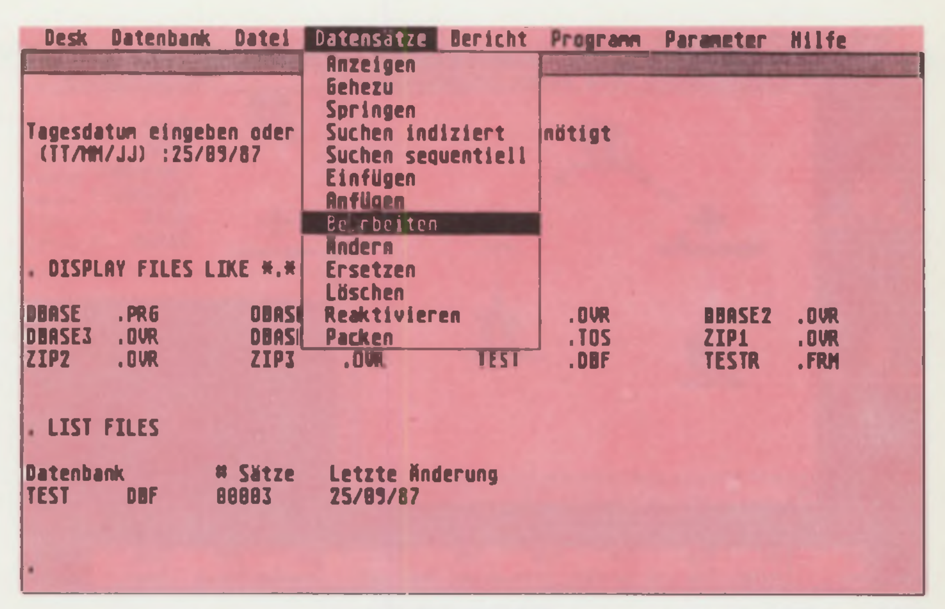


Bild 1: dBase II mit GEM-Oberfläche

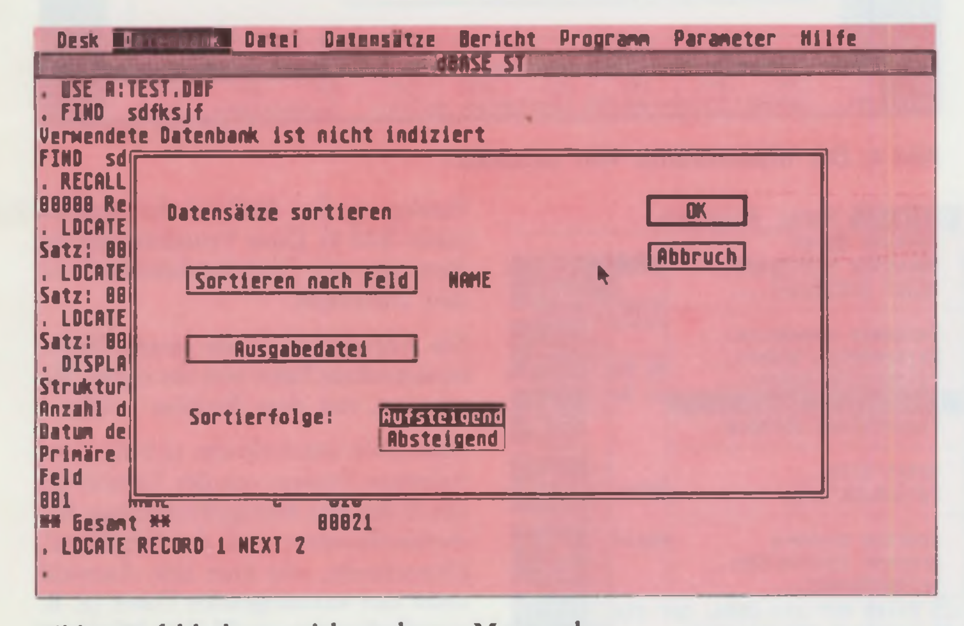


Bild 2: Befehle lassen sich auch per Maus geben

Beginnen wir mit 'dBase II unter GEM-TOS'

Das von vielen anderen Computern (in erster Linie PC-kompatiblen Rechnern) bekannte dBase II, das schon fast einen Industriestandard für Datenbanken auf PCs darstellt, wurde von Markt & Technik an den Atari ST angepaßt (Bild 1). Dazu wurde das Programm auf der neuen Maschine kompiliert und kommt daher jetzt ohne den zuvor erforderlichen CP/M-Emulator aus. Zusätzlich wurde eine GEM-Oberfläche geschaffen, die es ermöglicht, die meisten dBase-Befehle mit der Maus aufzurufen. Die GEM-Oberfläche setzt die grafisch gegebenen Befehle in englische dBase-Instruktionen um. Da die Möglichkeiten Befehle unter GEM aufzurufen, fast wie bei Superbase sehr weitreichend sind, kann man auch ohne Kenntnis der dBase-Programmiersprache schnell und einfach Datenbanken aufbauen und pflegen.

Die Programmiersprache dBase

Zu dBase II gehört standardmäßig die Möglichkeit, eigene Programme zu erstellen. Neben der Erstellung von Programmen, ist die Sprache auch dazu gedacht, in einem Direktmodus Instruktionen an das System zu geben. Man kann sich auf diese Weise schnell bestimmte Datensätze anzeigen lassen etc. Allerdings lassen sich die meisten der direkt ansprechbaren Funktionen auch über GEM erreichen. Interessant wird diese Sprache erst, wenn man sie in Programmen einsetzt. Sie bietet all das, was andere Programmiersprachen auch bieten, also Variablen, Prozeduren, Programmablaufkontrollbefehle etc.; zusätzlich verfügt sie über diverse Funktionen für die Handhabung von dBase-Dateien. Die Sprache ist so

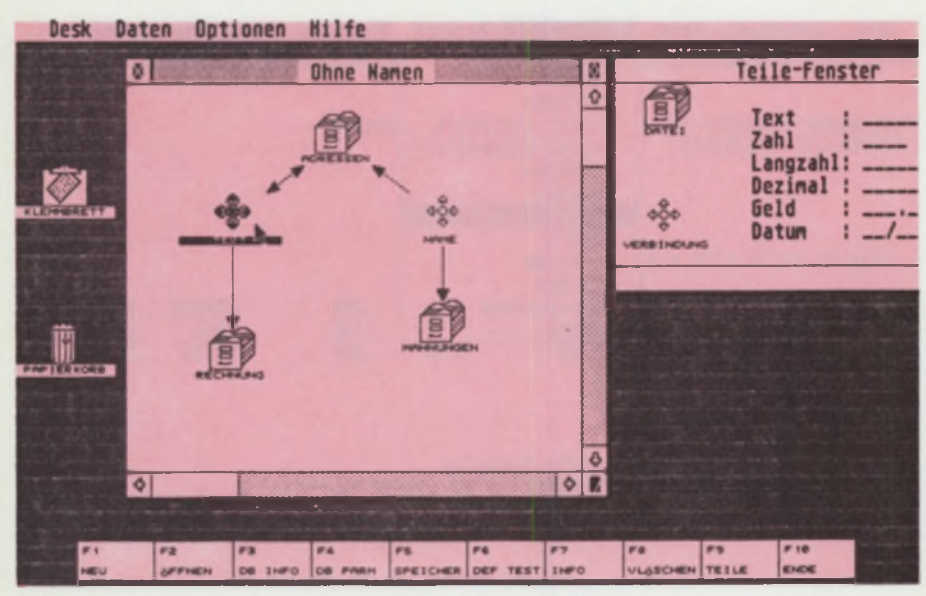


Bild 3: ADI-Init arbeitet nach WYSIWYG-Prinzip

mächtig, daß es für dBase bereits fertige Programme, wie Finanzbuchhaltungen etc. gibt.

All diese Programme können die Datenbankfunktionen von dBase nutzen. Normalerweise ist es am sinnvollsten, sich ein spezielles dBase-Programm zu schreiben; aber auch die direkte Abfrage und Pflege der Datenbank mit dBase ist möglich. Diese direkte Abfrage mit dBase-Befehlen, wurde in der ST-Version dadurch vereinfacht, daß man bestimmte Bedingungen für die Suche/das Sortieren in der Datenbank mit der Maus bestimmen kann (Bild 2).

Das neue 'Adimens ST V2.1'

Hatten wir bereits in der ST Computer 1/87 die damalige Version von Adimens ST ausführlich vorgestellt, so wollen wir hier auf die Neuigkeiten der Version 2.1 eingehen.

Auf der Systemdiskette der neuesten Adimensversion 2.1 (Bild 3) befindet sich ein unter GEM arbeitendes Programm namens INIT, das die Erstellung neuer Datenbanken extrem vereinfacht. Nach dem Top-Down-Prinzip kann hiermit leicht, mit grafischer GEM-Unterstützung der Aufbau der Datenbank festgelegt werden. Bild 3 zeigt die oberste Ebene beim Entwurf der Datenbank. Hier wird die Verknüpfung der logischen Dateien festgelegt. Der nächste Schritt ist dann die Festlegung des Aufbaus der Dateien (siehe Bild 4). Diese Visualisierung des Vorgangs des Datenbankentwurfs ist sehr vorteilhaft.

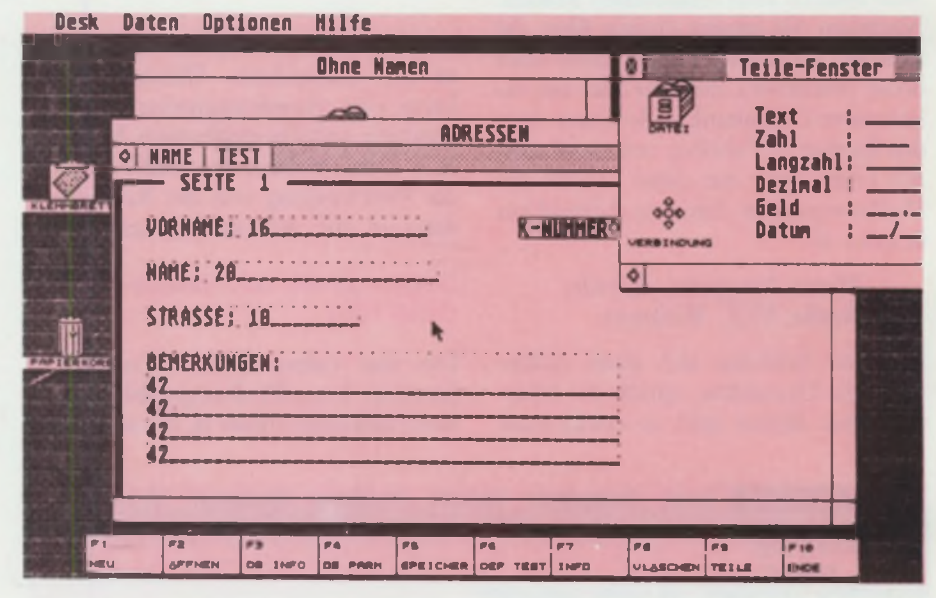


Bild 4: Der Maskeneditor von Adimens

Die Entwicklung einer neuen Datenbankstruktur kann von der ersten Minute an, mit dem Rechner erfolgen.

Ansonsten unterscheidet sich die neue Adimens Version von der Version 1.6 durch die neu hinzugekommenen Rechenfunktionen und das Clipboard. Ebensoleicht, wie man eine Auswahl unter den anzuzeigenden Daten (z. B. nur alle Kunden aus PLZ 4) definieren kann, läßt sich nun auch eine Berechnung vornehmen. So kann beispielsweise der Gesamtpreis aus der Anzahl, dem Stückpreis und der Mehrwertsteuer auf Knopfdruck berechnet werden.

Bild 5: Das Adimens 2.1 Hauptprogramm

Das Clipboard von Adimens 2.1 dient der Zwischenablage von Daten. Da die dort abgelegten Daten im RAM-Speicher gehalten werden, kann auf sie extrem schnell zugegriffen werden.

Insgesamt macht ADIMENS einen hervorragenden Eindruck, der in erster Linie auch durch die durchdachte GEM-Oberfläche zustandekommt (Bild 5). Das Programm bietet auch ohne Programmierung Flexibilität und einfachste Bedienung, wie sie andere Systeme selbst mit aufwendigen Programmen nicht erreichen. Mittlerweile gibt es für Adimens auch fertige Datenbanken zu kaufen. Dann braucht man sich nicht einmal mehr um den Aufbau (die Dateien und ihre Verknüpfung) Gedanken machen.

In Vorbereitung: ADI Talk

So wie in dBase eine Programmiersprache existiert, wird es (Mitte November) auch eine Programmiersprache für Adimens ST geben. Die Sprache ist nach Angaben von ADI dBase-kompatibel. Wir konnten uns bereits bei einer Vorversion von der Sprache überzeugen. Es stellt kein Problem dar, anhand der Standardliteratur Programme für ADI Talk zu erstellen. Schade ist nur, daß bei Verwendung dieser Sprache die GEM-Funktionen, wie das Adimens Exec.Prg sie unterstützt, verlorengehen. ADI-Talk wird für den Atari ST DM 189,– kosten und ermöglicht es, spezielle Anwendungen zu programmieren, für die das Adimens Programm zu flexibel (Fehlbedienung) oder zu unflexibel ist (eigene Programme wie z. B. eine FiBu).

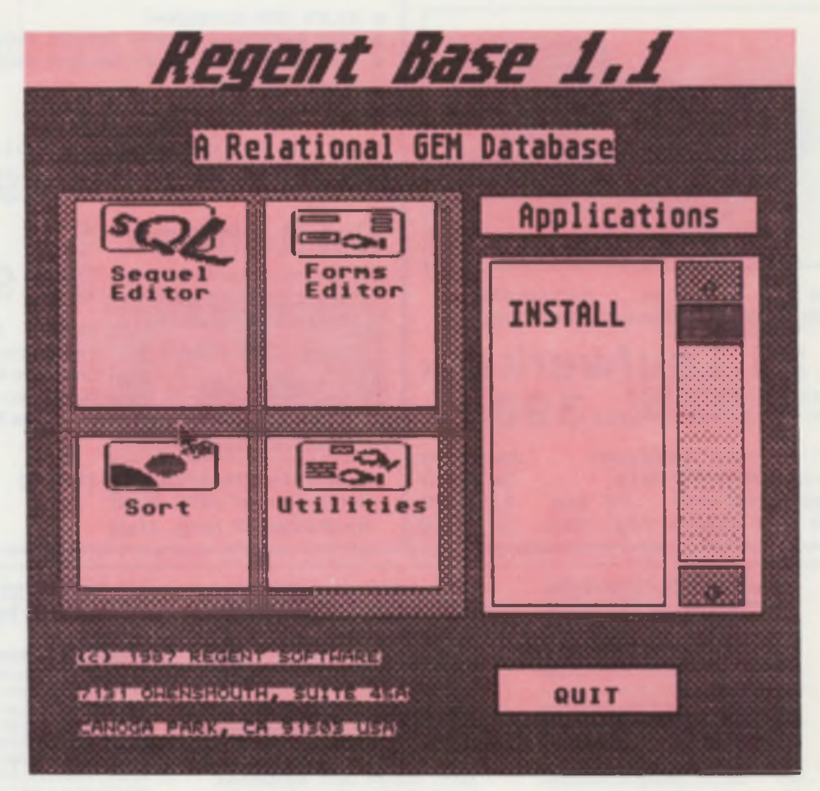


Bild 6: Hauptmenü RegentBase

Für ernsthaftere Programmierer gibt es die Adimens Datenbankfunktionen auch als Bibliothek, zur Einbindung in eigene C-Programme, zu kaufen. Bei Verwendung dieser Bibliothek können dann auch wieder die GEM-Funktionen genutzt werden. Das Adimens Exec, also das eigentliche Adimens ST wurde auch mit dieser Bibliothek entwickelt, die sich ADI C-Prog nennt und ca. DM 2000,– inklusive Lizenzgebühr kostet.

'RegentBase'

In Deutschland ist die amerikanische Firma Regent Software leider noch nicht sehr bekannt, dies allerdings zu unrecht. Regentbase besteht aus 5 Teilprogrammen, die durch ein Menüprogramm (Bild 6) aufgerufen werden. Nach der Erstellung einer Datenbank hält man sich im allgemeinen nur noch im Programm BASE.PRG auf. Dieses Programmodul führt selbstgeschriebene Regent Base-Programme aus. Doch soweit ist es noch nicht. Angenommen, man möchte keines der mitgelieferten Beispielprogramme benutzen, so muß zuerst eine Datenbank angelegt werden. Eine Datenbank kann z. B. mit einem mitgelieferten Utility-Programm definiert werden. Die Utility unterstützt die Mausbenutzung, so daß die Definition sehr schnell vorzunehmen ist. Hat man sich eine solche Datei geschaffen, so kann sie aus dem Hauptprogramm BASE bearbeitet werden. Hierfür dient die Sprache SQL, doch dazu später mehr.

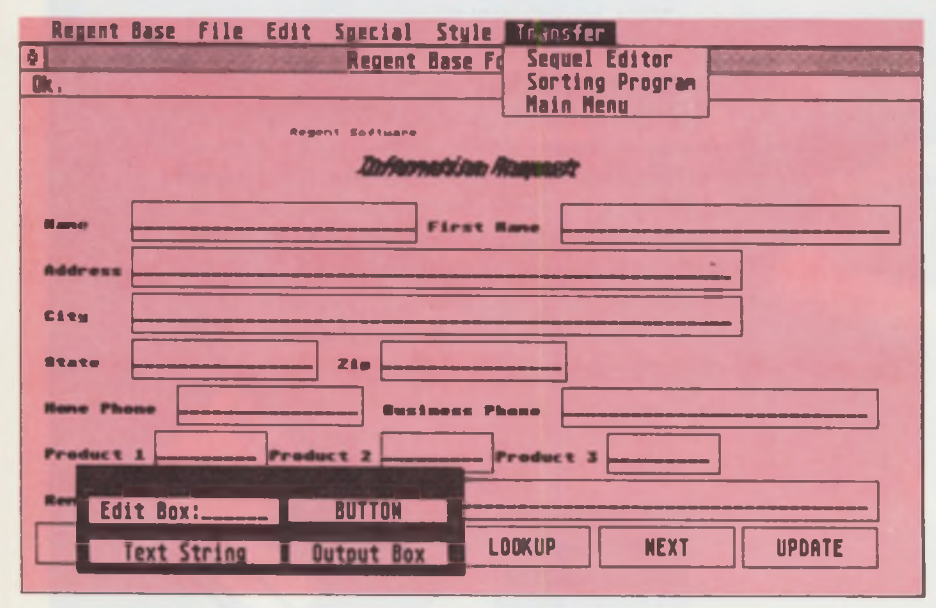


Bild 7: ...und der Masken- und Programmeditor von RegentBase

Zunächst gibt es da noch eine andere Möglichkeit: Man erstellt mit dem mitgelieferten Editor (Bild 7) ein Programm, daß eine Datenbank anlegt und die Pflege der Daten ermöglicht. Man kann komfortabel eine Eingabemaske generieren, die dann der Endbe-

nutzer verwendet. Hierfür bietet RegentBase auch Buttons an, so daß die erstellte Eingabemaske später auch mit der Maus bedient werden kann. Hat man diese Maske erstellt, so kann man sich zu jedem Feld eine Kennummer angeben lassen und im eigenen Programm leicht über diese Kennummer auf die eingegebenen Daten zugreifen. Wenn man im Programm schreibt %Kennummer, so wird hier automatisch mit dem gerade aktuellen Inhalt des Feldes gearbeitet. Über die Funktion USER$ kann das eigene Programm auch Mausklicks entgegennehmen.

Doch nun zur Programmiersprache selbst. Hier hat sich der Programmierer nicht die 'Mühe gemacht' eine eigene Sprache zu schaffen, sondern einen Standard integriert: Die in Amerika genormte Abfragesprache SQL.

Was sind Abfragesprachen?

Unter Abfragesprachen versteht man spezielle Sprachen, die nur dem Datenbanknutzer dienen. Mit diesen Sprachen kann man einfach definieren, welche Bedingungen die gesuchten Daten zu erfüllen haben. Zum Beispiel möchte man alle Kunden haben, die im Postleitzahlgebiet 1 wohnen, Atari ST Besitzer sind und mehr als 1000 DM im Monat verdienen, kennenlernen. Dafür möchte man kein langes Programm schreiben, sondern die Bedingungen möglichst ähnlich der menschlichen Sprache eingeben. Solange hierfür keine Systeme mit künstlicher Intelligenz allgemeine Verbreitung finden, verwendet man Abfragesprachen.

SQL oder Sequel

SQL (Structered Query Language = Strukturierte Abfrage Sprache), früher auch unter dem Namen Sequel bekannt, ist eine solche Abfragesprache speziell für relationale Datenbanken. Sie ermöglicht es dem Benutzer nach beliebigen Kriterien die Daten durchsuchen zu lassen. Dabei lassen sich alle Möglichkeiten relationaler Systeme nutzen. Dieser Tatsache ist es wohl auch zu verdanken, daß sich diese Sprache im Großrechnerbereich durchgesetzt hat. Sehr viele große Datenbanksysteme verwenden als Abfragesprache SQL. Regentbase bietet diese Sprache aus dem Profibereich erstmals auf einem PC-Programm für den ST. In der ST-Version stehen nicht nur diese SQL-Standardbefehle zur Verfügung, sondern auch die Kontrollstrukturen, die diese Sprache bietet. So ist es möglich komplette Programm emit Eingaben, Ausgaben, Reportfunktionen etc. einfach zu implementieren. Alle diese Möglichkeiten werden durch mächtige Befehle unterstützt. Die Datenbankfunktionen machten einen ausgereiften Eindruck. Zu bemängeln ist nur, das nicht alle SQL-Befehle integriert wurden. Außerdem traten mit GEM einige geringe Probleme auf. So habe ich es geschafft, daß die Ereignisse falsch abgearbeitet wurden. Selbst nach dem Verlassen des Programmes wurden alle Ereignisse verzögert ausgeführt. Nach ca. 15 Tastendrücken wird die 15 Tasten zuvor betätigte Taste ausgeführt. Ebenso bei Mausklicks. Dies führte zu einem nicht mehr überblickbarem Chaos.

'Superbase'

Von Precision Software kommt der vierte Vertreter der Datenbanken. Das wie Adimens oder dBase schon etwas ältere Programm (alt heißt nicht schlecht!) wurde an den Atari ST angepaßt. Es ähnelt in seiner Anpassung sehr stark der ebenfalls erhältlichen Amiga-Version (Bild 8). Unter dem Fenster befinden sich Tasten, die an ein Bandgerät erinnern. Hiermit kann man entsprechend einem Schlüsselfeld vor/zurückspulen, stoppen und Filterfunktionen aufrufen. Mit dem Filter kann die Anzeige der Datensätze an bestimmte Bedingungen geknüpft werden. Dabei lassen sich die Kriterien in verschiedenen Formularen durch einfaches Anklicken auswählen.

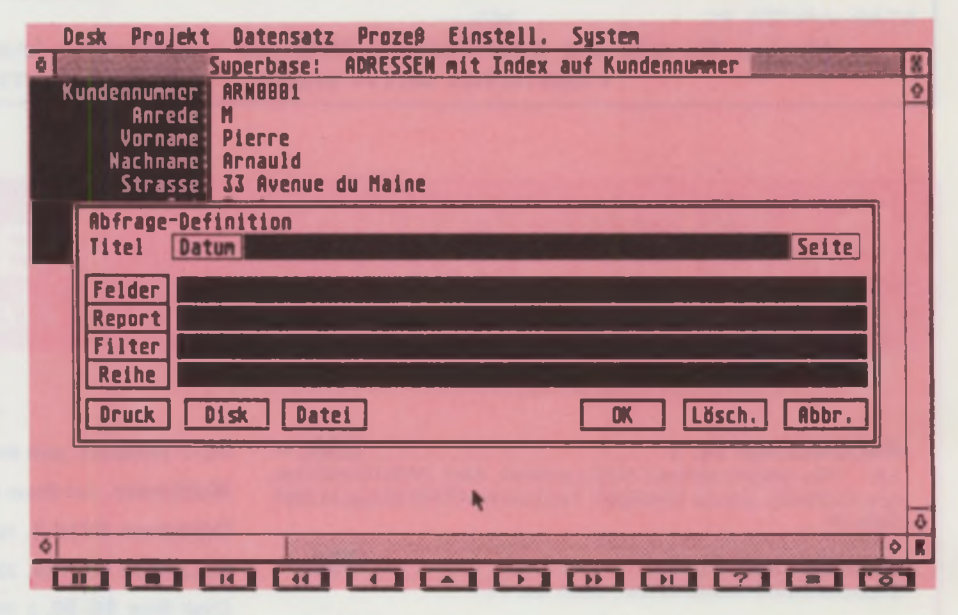


Bild 8: Diese Abfrage-Definition kann man allein mit der Maus durch Anklicken der Buttons ausfüllen

Doch zunächst einmal muß der Aufbau der Datenbank definiert werden. Superbase ist hier recht flexibel. Neben den üblichen Datentypen innerhalb von Dateien erlaubt Superbase auch eine Eingabeüberprüfung. Man kann vordefinieren, welche Werte (Wertebereich) ein Feld haben darf, und Superbase verhindert dann später bei der Benutzung der Datenbank Fehleingaben durch den Benutzer. Ebenso lassen sich Formeln vordefinieren, so daß bestimmte Datensatzelemente automatisch aus den Inhalten der anderen errechnet werden. Schade bei all diesen Definitionen ist, daß nur eine sehr kurze Zeile für die Formeln zur Verfügung steht. Hier hätte man wirklich ein Scrolling spendieren können.

Auch Grafiken...

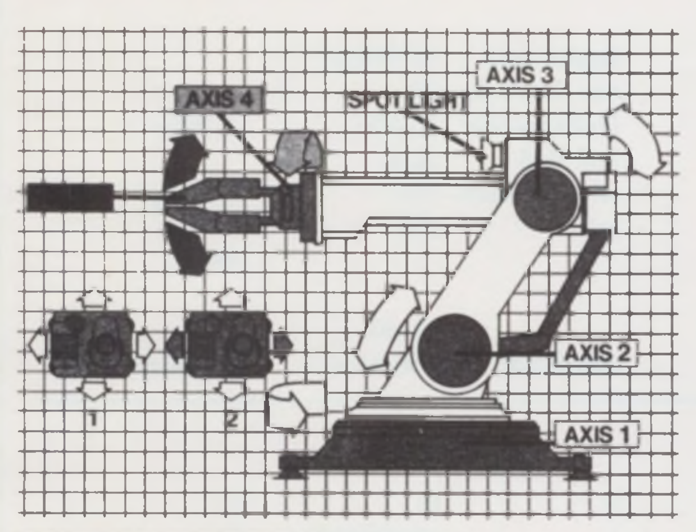
...lassen sich mit Superbase verwalten. Es ist das einzige Programm, daß in der Lage ist, innerhalb einer Datenbank einen Bezug zu einer externen Datei, wie z. B. Bildern (Bild 9) herzustellen. So könnten Sie zum Beispiel zu allen Geschäftsfreunden digitalisierte Bilder speichern. Oder Sie speichern zu einem Angebot die entsprechende Konstruktionsskizze.

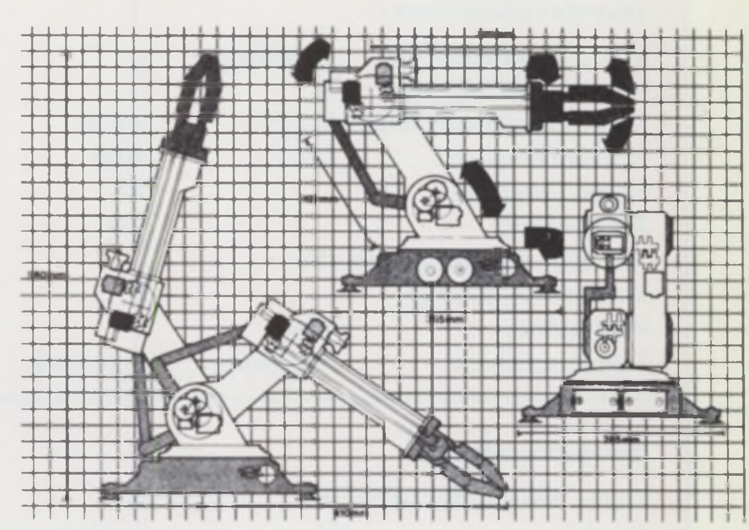
Von anderen Rechnern kennt man bei Superbase die Möglichkeit, ebenfalls eigene Programme für Superbase zu schreiben. Bei Superbase ST scheinen diese Möglichkeiten auch implementiert worden sein, nur fehlt hier eine Funktion, die Programme zu laden und zu starten. Das Handbuch gibt darin keine Auskunft. Es gibt nur Aus-

ZUKUNFT

NEU

schon heute hautnah erleben.
Endlich ein semiprofessioneller 5achsiger Roboterarm für jedermann erschwinglich.
Anschlußfertig mit Software für ST / AMIGA / PC / AT







- für Schulungszwecke
- für Lehr- und Demonstrationszwecke
- zum Spielen – für Ihr Hobby

Technische Daten

- 5 Achsen
- zugkräftige Elektromotoren
- Gewicht: ca. 2,5 kg
- Farbe: industriegelb
- Höhe ges.: ca. 54 cm

Interface und Software für:

- ATARI ST (GEM Software, Teach Funktion)
- Commodore AMIGA
- PC / AT kompatibel

- einfacher Anschluß über unser Interface an den Druckerport
- auch ohne Interface über 2 Joysticks zu steuern

- aus allen Programmiersprachen sehr einfach anzusteuern



Roboterarm	**DM 124,50**	**2 Joysticks**	**DM 25,—**
Interface & Software	**DM 144,50**	**Batteriesatz**	**DM 10,—**
Gesamt	**DM 269,—**	**Spezialnetzteil**	**DM 99,—**

G DATA
Siemensstr. 16 · 4630 Bochum 1
TELEFONISCHE BESTELLUNGEN: 023 25 / 6 08 97

Schweiz: Senn Computer AG
Langstr. 21 · CH-8021 Zürich

Bestell-Coupon
Bitte senden Sie mir:

☐ für ST ☐ für AMIGA ☐ für PC ☐ für AT
zzgl. DM 5,— Versandkosten
☐ per Nachnahme ☐ Verrechnungsscheck liegt bei

Name
Straße
Ort

Infobox Datenbanksysteme

Die Merkmale der verschiedenen relationalen Datenbankprogramme im direkten Vergleich:

Produktname:	Adimens ST	dBase ST	RegentBase	StandardBase[2]	Superbase ST
Datenbankfunktionen:					
Max. Datenbankgröße	16 MByte		Begrenzung	63 MByte[1]	
Max. Anzahl log. Dat	32		nur durch die	32	
Max. Datensatzzahl/Datei	65 535	65 535	Diskkapazität	16 Mio.	16 Mio.
Zeichen/Datensatz	8191	1000	und Bild-	32 K	
Felder/Datensatz		32	schirmmasken-	2000	beliebig
Zeichen/Feld		254	größe	253	
Maximale Anzahl Indizes	191				999
Max. Indexschl.-länge		99			
Verwendetes	Modif.			Modif.	
Zugriffsverfahren	B★-Bäume	unbekannt	unbekannt	B★-Bäume	unbekannt
Benutzeroberfläche:					
Eigene Bildschirmmasken	ja	ja	ja	ja	ja
Anzahl	63	prgabhängig	1	prgabhängig	1
Rechenfunktionen:					
Genauigkeit in Stellen	23	10	wählbar	15	wählbar
Reportfunktionen:					
Frei editierb. Format	nein[4]	ja	ja	prgabhängig	ja
Datenimport und -exportfunktionen:					
Im-/Export von ASCII	ja	ja	ja	ja	ja
Anderes Format	–	–	dBase III	dBase III	–
Zusatzfunktionen:					
Bildverarbeitung	nein	nein	nein	nein	ja
Programmierbar	optional	ja	ja	ja	siehe Text
Verwendete Sprache	DBase II	DBase II	SQL (Sequel)	DBase III	Superbase
Paßwortschutz	ja	prgabhängig[3]	prgabhängig[3]	prgabhängig[3]	ja
Entwickler:	ADI	Ashton Tate	Regent Soft.	Knupe GmbH	Precision Soft.
Vertrieb:	Atari GmbH 6096 Raunheim	Markt & Technik 8013 Haar	H. Haase 4300 Essen	Knupe GmbH 4600 Dortmund	Markt & Technik 8013 Haar
Preis DM:	199,–	348,–	249,–	698,–	249,–

[1] GEMDOS unterstützt nur 16 MByte

[2] Vorvorversion

[3] Kein interner Paßwortschutz, also leicht überwindbar

[4] Dafür ASCII-Export zu Word-Plus

kunft über die reservierten, aber nicht benutzbaren Befehle.

Wie steht's mit der Paperware – entscheidend ist das Handbuch

Für so komplexe Programme wie Datenbanksysteme sind gute Handbücher unerläßlich. So werden bei den getesteten Produkten ohne Ausnahme deutsche Handbücher mitgeliefert. Dabei gehört das dickste Handbuch zu dBase. Die vielfältigen Funktionen und die eingebaute Programmiersprache erfordern ein umfangreiches Nachschlagewerk, das sich in einen atarispezifischen und einen allgemeinen dBase-Teil gliedert. Der atarispezifische Teil beschreibt die Bedienung der neu integrierten GEM-Oberfläche. Den zweiten Teil des Handbuches bildet das Originalhandbuch zu dBase II, wie es in der PC-Version vertrieben wird. Da dieses Buch sehr umfassend ist und alle Funktionen detailiert beschreibt, ist es nicht unbedingt ein Nachteil, daß das Originalhandbuch verwandt wird. Nur die Teile zur Anpassung und Installation des Systems, die beim Atari überflüssig sind, hätte man sich besser sparen können.

Zu Adimens 2.1 gehört nun natürlich auch eine neue Anleitung, die die erweiterten Fähigkeiten beschreibt. Besonders hervorzuheben an dieser Anleitung, ist das ausführliche Tutorial, daß besonders für Einsteiger sehr hilfreich ist. Hier lernt man auf einfache Weise den Umgang mit dem Programm. Man lernt eine Datenbank zu erstellen und mit der mitgelieferten Beispieldatenbank auch schnell die Handhabung und Pflege fertiger Datenbanken. Das Handbuch erscheint gut durchdacht, nur fehlte bei der uns vorliegenden Vorversion noch ein Stichwortverzeichnis. Es ist zu hoffen, das dieser Mangel in den endgültig ausgelieferten Versionen behoben sein wird.

Das Handbuch von Regentbase ist gut gegliedert. Es erläutert erst die Bedienung der Programmodule und dann das Erstellen eigener Programme mit der Sprache SQL. Für SQL ist auch ein alphabetisch sortiertes Verzeichnis mit der genauen Definition angefügt. Für Programmierer wird auch das Format der Dateien erläutert.

Markt und Technik hat das Handbuch zu Superbase geschrieben. Neben der genauen Beschreibung der Bedienungsabläufe enthält das Handbuch auch locker eingestreute Übungen zum Erlernen der Programmbedienung. Durch die mitgelieferten Beispieldateien, auf die auch im Handbuch eingegangen wird lernt man schnell die Benutzung dieses Programmes.

Fazit

Bei allen vier Programmen handelt es sich um leistungsfähige Datenbankprogramme, die in ihren Funktionen weit über die Möglichkeiten vieler einfacherer Dateiverwaltungen herausgehen. Alle Programme wurden ursprünglich auf einem anderen Computer implementiert. Die Anpassungen an den Atari ST und sein GEM sind verschieden gut gelungen.

Mit dem Preis von DM 199,– steht Adimens angesichts seiner Leistungsfähigkeit und Ausgereiftheit relativ konkurrenzlos dar. Besonders einfache Dateiverwaltungen sehen angesichts dieses Preises sehr alt aus. Wem es nicht so sehr auf die Preisdifferenz ankommt, sollte aber auch die anderen Programme in seine engere Wahl ziehen. Teilweise werden hier interessante Ideen geboten, so daß jedes Programm seine individuellen Vorteile und Einsatzgebiete hat. dBase ist zum Beispiel dort interessant, wo bereits fertige dBase-Programme verfügbar sind. Außerdem existiert zu diesem Programm die meiste Sekundärliteratur und das größte Schulungsangebot (zur PC Version). Von diesem Angebot kann allerdings auch der ADI-Talk-Benutzer Vorteile haben, da sich ADI-Talk nur wenig von dBase unterscheidet. Wer Erfahrungen mit SQL hat, oder diese sammeln will, dem ist RegentBase zu empfehlen und Superbase schließlich bietet einzig die Einbindung von Grafik. Ansonsten sei auch noch auf die Tabelle verwiesen, in der alle Programme einander gegenübergestellt werden. Hier werden die Begrenzungen, Standard- und Sonderfunktionen verglichen.

Kurz vor Redaktionsschluß...

...erreichte uns eine Demoversion von **STandard Base,** einem DBase III kompatiblen Programm. Es ist eine Umsetzung eines Mac-Programmes für den Atari ST. Das Programm (Bild 10) kann unter DBase III entwickelte Anwendungen ausführen und unterstützt das Originalformat, so daß Dateien vom PC übernommen werden können. DBase III ist das Nachfolgeprogramm von DBase II und bietet Erweiterungen in Bezug auf die Größe der verwaltbaren Dateien, die Funktionen und die Bearbeitungsgeschwindigkeit. Das Programm STandard Base wurde wie Adimens komplett in Deutschland entwickelt. Leider konnten wir bis zum Redaktionsschluß nur eine sehr frühe Vorversion testen. Das Konzept des Programmes entspricht in etwa der dBase-Version von Markt & Technik: Auf ein textorientiertes Grundsystem wurde eine grafische Oberfläche aufgesetzt, die es ermöglicht, Befehlsfolgen einfach mit der Maus aufzurufen. Vom Vorbild soll sich STandard Base nach dem Willen der Entwickler, vor allem auch durch die hohe Zugriffsgeschwindigkeit abheben.

(JW)

Bild 9: Superbase kann auch Bilder speichern

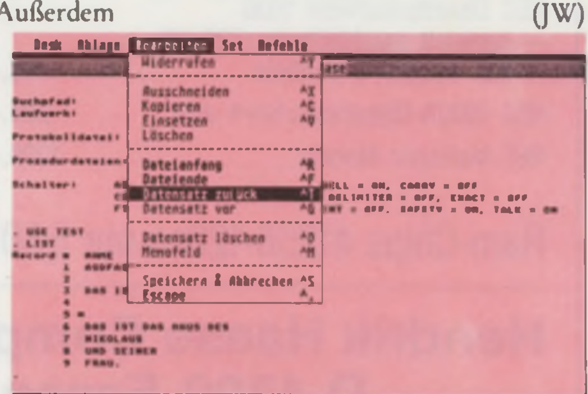


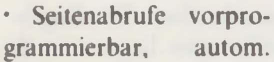
Bild 10: StandardBase ist dBase III kompatibel

SOFTWARE NEWS



I ST Speeder
· Beschleunigt das Arbeiten mit Diskette bis zu Faktor 10 !· Arbeitet mit intelligentem Cache-Memory · Ein wirkliches Muß für den professionellen Anwender · Arbeitet mit neuem und altem ROM, S/W und Farbe · Wird mit deutscher Anleitung geliefert
Preis: DM 98,-*

BTX ST/ALADIN
· Endlich BTX auf dem ST ! · Telex über BTX · Seitenabrufe vorprogrammierbar, autom. Einloggen · Einfügen von Grafik/Texten u.v.a.m. · Erfordert ALADIN (Software oder Hardware) und S/W-Monitor · Upgrades für TOS lieferbar 12/87
Preis: DM 398,-*





MusiX32
· Der Standard auf dem ATARI ST · Ermöglicht Ausdruck der Noten auf einem Drucker · Einbindung in C-, Pascal- und GFA-Basic-Programme möglich · Arbeitet auf S/W-Systemen · Wird mit ausf. deutscher Anleitung geliefert
Preis: DM 89,-*

LisPas II ST/LisPas Tutor
· Der preiswerte Einstieg in Künstliche Intelligenz · Vollständiger LISP-Interpreter · GEM-Oberfläche · Lehrgang erhältlich (LisPas Tutor) · Kein Kopierschutz
Preis: LisPas II ST **DM 198,-***
Handbuch einzeln: **DM 49.80***
LisPas Tutor: **DM 148,-***

Dizzy Wizard
· Das Spiel, das keinen Blitter braucht · 100 verschiedene Levels · bis zu drei Spieler · läuft auf S/W und Farbe · Test in ST 9/87 · Digisound
Preis: DM 98,-*

Programmierer für ATARI ST/AMIGA gesucht – Schreiben Sie uns !



TrashHeap
· Das ultimative 3D-Weltraumspiel · Wurde schon im Fernsehen und Radio vorgestellt · Wird mit 3D-Brille und Anleitung (deutsch) geliefert · läuft auf S/W und Farbe · Digisound
Preis: DM 98,-*



Intelligent Spooler ST
· Intelligenter Drucker-Spooler · Reihenfolge der Texte änderbar · Speichert bis zu 16 MB-Daten · Anpassbar an jeden Drucker · läuft auf S/W und Farbe
Preis: DM 98,-*

Crypt-it – Daten und Programmverschlüsselung DM 98,- · **Lock-it** – Der Kopierschutz für Anspruchsvolle DM 298,- · **BinLook** – Die Bildergalerie unter DEGAS für den ATARI ST DM 79,-

* Alle Preise sind empfohlene Verkaufspreise ! Händleranfragen erwünscht ! Alle Programme werden auf einseitigen Disketten ausgeliefert !

Irrtum und Änderungen vorbehalten.

ANRUF GENÜGT:
069/61 40 46

Bestellcoupon
☐ Bitte senden Sie mir kostenloses Informationsmaterial (DM 1,40 Rückporto liegt bei)
☐ Hiermit bestelle ich

zzgl. DM 6,- Versandkosten (bei Vorauskasse)
☐ per Nachnahme ☐ Verrechnungsscheck liegt bei

Name
Straße
Ort

ST 11/87

TOMMY SOFTWARE

Gutzkowstr. 35, D-6000 Frankfurt/M 70, Tel./BTX: 069/61 40 46, Telex über BTX, KEIN LADENVERKAUF

Schweiz: Senn Computer AG, Langstr. 21, CH-8021 Zürich, Tel.: 01/24 17 37 3 Österreich: Computer-Studio Wehsner GmbH, Paniglgasse 18-20, A-1040 Wien, Tel.: 0222/65 78 08

Der Krieg der Kerne

Ein völlig neuartiges Computerspiel



Die Bezeichnung „Spiel" ist etwas verwirrend: Weder benötigt man einen Joystick, noch gibt es Gewehrsalven oder UFOs, die über den flackernden Bildschirm sausen. Freilich kann man den „Krieg der Kerne" auch kaum als Anwenderprogramm bezeichnen, denn die spielerische Komponente ist unübersehbar: Ein Kampf, aus dem Sieger und Unterlegene hervorgehen.

Das Schlachtfeld, auf dem es zu siegen gilt, ist das Innere des Computers. Genauer gesagt: Ein vorher festzulegender Speicherbereich von zum Beispiel 8000 Bytes Länge.

Zwei Computerprogramme sind die Gegner, die sich in dieser Arena den Garaus zu machen versuchen. Sie sind in einer künstlichen, assembleráhnlichen Sprache geschrieben. Die gegnerische Position und Strategie ist völlig unbekannt. Fest steht nur, daß der Feind alles daransetzt, seinen Gegner durch Einpflanzen eines ungültigen Befehls zum Absturz zu bringen.

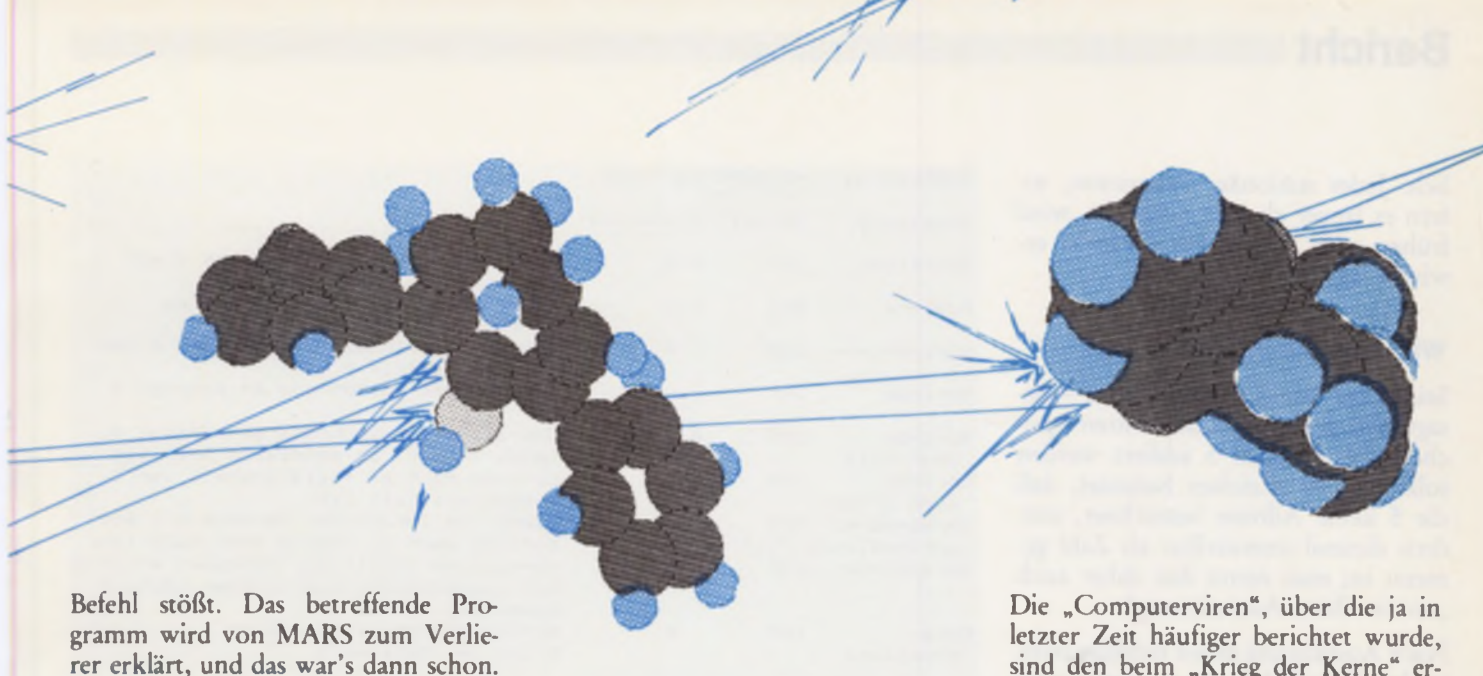
Die Rolle des Spielers beim „Krieg der Kerne" besteht allein darin, ein möglichst raffiniertes Programm in der bereits erwähnten Sprache zu erstellen, die von ihrem Erfinder REDCODE genannt wurde. Dieses Programm muß in der Lage sein, ein gegnerisches Programm aufzuspüren und es durch „Beschießen" mit ungültigen Befehlen außer Gefecht zu setzen. Natürlich kann es auch einen anderen Weg einschlagen, indem es eine starke Verteidigung gegen gegnerischen Beschluß aufbaut und eigene Wunden, also Treffer, gegebenenfalls sogar selbst repariert. Der Phantasie des Programmierers sind hierbei kaum Grenzen gesetzt. Denkbar sind auch Programme, die sich todbringenden Treffern entziehen, indem sie sich ziellos im Speicher hin- und herkopieren und dabei womöglich selbst den Gegner ins Visier nehmen oder ihn gar überrollen. Hat so ein Programm noch heute viele Kämpfe überstanden, so kann es morgen schon von einem stärkeren Gegner in seine Schranken verwiesen werden.

Hat man also das Programm erstellt und für tauglich befunden, so gibt es nur noch eines zu tun: Es muß zusammen mit dem gegnerischen Programm in die Arena geschickt werden, um sich zu bewähren. Der weitere Ablauf entzieht sich jeglicher Einflußnahme durch den Programmierer: Der Kampf findet in aller Stille statt. Kein Gewehrdonner, kein Schlachtengetümmel läßt merken, daß im Inneren des Computers ein mitunter recht heftiges Gefecht tobt. Erst die trockene Textzeile „Sieger ist Programm X" verkündet nach oft überraschend kurzer Zeit Ende und Ergebnis des Kampfes.

Bevor wir uns jedoch den Details widmen, muß erwähnt werden, daß die Idee zum „Krieg der Kerne" von A. K. Dewdney stammt, der sie bereits vor einigen Jahren in zwei Artikeln der Zeitschrift „Spektrum der Wissenschaft" (Hefte 8/84 und 5/85) publizierte. Dort findet man zwar viele Einzelheiten, Tips und Anregungen, jedoch keine Inplementierung für einen bestimmten Computertyp.

Weil nun das „Spektrum der Wissenschaft" nicht gerade jedermanns Frühstückslektüre ist, doch die Idee von A. K. Dewdney insgesamt sehr interessant erscheint, haben wir uns zu zweit sofort daran gemacht, eine Version für den Atari ST zu schreiben. Das Ergebnis dieser Bemühungen liegt nun als Public-Domain-Programm vor, das auf Diskette 57 enthalten ist.

Der Leser wird sich hier wohl fragen, wie es denn möglich sei, zwei Programme gleichzeitig ablaufen zu lassen. Selbst der smarteste Computer kann doch gewöhnlich nur einen Befehl auf einmal abarbeiten. Die Lösung lautet „Time sharing". Dies bedeutet lediglich, daß sich beide Programme die Computer-Zeit „teilen". In der Praxis sorgt ein übergeordnetes Programm namens MARS (Memory Array Redcode Simulator) dafür, daß erst ein Befehl des einen Kampfprogramms abgearbeitet wird, dann ein Befehl des anderen Programms und abwechselnd immer so weiter. Dieser Vorgang endet, wenn MARS auf einen ungültigen



Befehl stößt. Das betreffende Programm wird von MARS zum Verlierer erklärt, und das war's dann schon.

Wie sieht nun solch ein REDCODE-Programm aus?

Zur Erläuterung sei hier das absolut kürzeste REDCODE-Programm vorgestellt. Es nennt sich KNIRPS und ist trotz seiner Winzigkeit ein durchaus ernstzunehmender Gegner im „Krieg der Kerne“:

```
MOV 0 1
```

KNIRPS tut nichts anderes, als sich bei jedem Aufruf um genau eine Speicherstelle weiterzubewegen. Das Mnemonik MOV bewirkt, daß der Inhalt, der durch die erste Zahl bezeichneten Speicherstelle an die, durch die zweite Zahl bestimmte, kopiert wird (MOVE: engl. „bewegen“). Die Zahlen geben dabei eine relative Adresse an, d. h. der Inhalt von (augenblickliche Adresse +0) – es handelt sich um den Befehl selbst – wird nach (augenblickliche Adresse +1) kopiert. Freunde der Assemblersprache haben es hier leichter, aber mit etwas Gewöhnung wird dies auch für Hochsprachenprogrammierer zu durchschauen sein.

KNIRPS bewegt sich also mit einer Geschwindigkeit von einer Speicherstelle pro MARS-Zyklus im Speicher vorwärts. Ohne ausreichende Gegenmaßnahmen würde er früher oder später in das gegnerische Programm hineinrennen und es einfach überrollen. Dies würde zwar nicht dazu führen, daß MARS beim Gegner auf einen ungültigen Befehl träfe – es fände lediglich eine Kopie von KNIRPS vor. Der Gegner ist jedoch „umgedreht“ worden, d. h. von nun an rennen zwei KNIRPSE durch den Speicher, ohne sich jemals zu kriegen oder sich Schaden zuzufügen. Das Spiel wird unentschieden enden.

Sollte einer unserer KNIRPSE dabei das obere Ende des Speicherplatzes erreichen, so läuft er einfach an dessen Anfang weiter, was uns zu einem besonderen Merkmal des „Schlachtfeldes“ führt: Dieser Speicherbereich ist als Ring zu verstehen. An seine höchste Adresse schließt die niedrigste Adresse unmittelbar an. Hat das Schlachtfeld zum Beispiel eine Größe von 8000 Bytes und würde ein Befehl, der sich auf Adresse 7900 befindet, die Adresse 200 ansprechen, so würde dies von MARS in die Adresse 100 übersetzt.

augenblickliche Adresse:
7900

z. B. MOV 0 200:
7900 + 200 = 8100

wird zu:
8100 mod 8000 = 100

Die „Computerviren“, über die ja in letzter Zeit häufiger berichtet wurde, sind den beim „Krieg der Kerne“ erstellten Programmen nicht unähnlich, denn ihre Funktion besteht ja auch meist darin, andere Programme zu zerstören oder unbrauchbar zu machen. Beim „Krieg der Kerne“ bleiben jedoch, durch die geschlossene Struktur der „Arena“, die Tierchen im Stall und können keinen Unfug mit dem Betriebssystem oder mit anderen Programmen anstellen.

Weitere Feinheiten der REDCODE-Programmierung seien anhand eines zweiten kurzen Kampfprogrammes namens GNOM erklärt (siehe Tabelle 1).

Um es gleich vorwegzunehmen: Der GNOM tut nichts anderes, als jede fünfte Speicherstelle mit einer 0 zu belegen. Die 0 stellt in REDCODE einen ungültigen Befehl dar. Sollte sich an ihrer Stelle vorher das feindliche Programm befunden haben, so wird es bei Abarbeitung dieser „Bombe“ unweigerlich abstürzen. Die Gefährlichkeit des GNOMS ist also offensicht-

Adresse	1. Zyklus	2. Zyklus	9. Zyklus
0	----------	----------	----------
1	ADD #5 3	ADD #5 3	ADD #5 3
2	MOV #0 @2	MOV #0 @2	MOV #0 @2
3	JMF -2	JMF -2	JMP -2
4	DAT -4	DAT 1	DAT 11
5	----------	--- 0 --	--- 0 --
6	----------	----------	----------
7	----------	----------	----------
8	----------	----------	----------
9	----------	----------	----------
10	----------	----------	--- 0 --
11	----------	----------	----------
12	----------	----------	----------
13	----------	----------	----------
14	----------	----------	----------
15	----------	----------	--- 0 --

Tabelle 1: klein aber aggressiv: der GNOM

lich. Jedes stationäre Programm, sofern es länger als vier Zeilen ist, wird früher oder später von GNOM erwischt werden.

Wie geht GNOM dabei vor?

Sein erster Befehl – ADD #5 3 – besagt, daß zum Inhalt der dritten Speicherstelle die Zahl 5 addiert werden soll. Das '#' Zeichen bedeutet, daß die 5 keine Adresse bezeichnet, sondern diesmal unmittelbar als Zahl gemeint ist; man nennt dies daher auch „unmittelbare Adressierung".

Nach Ausführung dieses Befehles steht in der Zeile DAT −4 nunmehr statt −4 der Wert 1 (−4+5=1).

In dem zweiten Befehl – MOV #0 @2 – lernen wir eine weitere Adressierungsart kennen: die indirekte Adressierung. Der Befehl bedeutet: Schreibe die Zahl 0 (unmittelbar!) an DIE ADRESSE, DIE IN DER, DURCH DIE INDIREKTE ADRESSE 2 BEZEICHNETEN SPEICHERSTELLE steht. MARS schaut also zuerst nach, welche Speicherstelle mit @2 gemeint ist: Es ist wiederum unser DAT-Befehl zwei Zeilen tiefer. Welche Zahl steht nun dort? Es ist noch von vorhin – die 1. Da der DAT-Befehl selbst auf Speicherstelle 4 steht, ist durch MOV #0 @2 die Speicherstelle 4+1=5 gemeint, also direkt hinter dem DAT-Befehl. Der MOV-Befehl soll demnach eine „Bombe" an die durch den DAT-Befehl bestimmte Speicherstelle legen.

Der dritte Befehl – JMP −2 – ist schnell erklärt: Er weist MARS an, zur Bearbeitung des nächsten Befehls zwei Schritte zurückzugehen, also wieder zum ADD-Befehl (JUMP: engl. „springe").

Um es zusammenzufassen: Der GNOM besteht aus einer Endlosschleife, in der erst der Inhalt einer bestimmten Speicherstelle um 5 erhöht wird, um dann an den, durch diese Speicherstelle bezeichneten Ort eine Bombe in Form eines ungültigen Befehles – der NULL – zu legen.

Wir haben nun alle drei Adressierungsarten, und zwar (s. Tabelle 3)

DIREKT

Das Argument gibt an, wieviele Speicherstellen ausgehend von der Position des gerade bearbeiteten Befehls – weiter gezählt werden muß, um die gewünschte Speicherstelle zu erhalten. DIREKTE Argumente haben kein Vorzeichen.

UNMITTELBAR

Das Argument bezeichnet keine Speicherstelle, sondern einen Wert, der unmittelbar als Zahl zu benutzen ist.

UNMITTELBARE Argumente sind durch ein '#' gekennzeichnet.

INDIREKT

Das Argument verweist – nach Art der direkten Adressierung – auf eine andere Speicherstelle. Der Inhalt dieser Speicherstelle wiederum kann nun als direkte Adresse aufgefaßt werden, d. h. es wird um den Betrag der dort stehenden Zahl weiter gezählt, um die durch das Indirekt-Argument bestimmte Speicherstelle zu erhalten.

INDIREKTE Argumente sind durch ein '@' gekennzeichnet.

Im übrigen gilt für alle Argumente, daß sie auch mit einem negativen Vorzeichen versehen werden können. Solche negativen Zahlen werden vom MARS sofort in ihr positives Komplement umgewandelt, denn in einem ringförmigen Speicher läuft die Subtraktion von z. B. 1 auf dasselbe hinaus wie die Addition von 8000−1, also 7999.

Noch ein Wort zum DAT-Befehl: Er hat eine ähnliche Funktion wie z. B. der DATA-Befehl in BASIC. Man kann mit ihm Daten oder Zahlen speichern. Man sollte sich jedoch merken, daß der DAT-Befehl keine ausführbare Anweisung darstellt: Falls MARS bei der Abarbeitung eines Programms auf einen solchen Befehl trifft, bedeutet dies für das betreffende Programm den Todesstoß.

Außerdem ist noch zu erwähnen, daß GNOMs „Bombardement" ebenfalls den Gesetzen des zirkulären Speichers unterliegt. Wenn seine Bomben nämlich das obere Ende des Speichers erreicht haben, beginnen sie wieder im unteren Ende. Dies hat interessanterweise zur Folge, daß – wenn die Zahl der Speicherstellen der „Arena" nicht durch 5 teilbar ist – der GNOM sich früher oder später „von vorn in den Rücken schießt". Das muß ihm erst mal jemand nachmachen...

Ist die Länge des Speichers jedoch genau durch 5 teilbar, dann schießt er stets haargenau an sich selbst vorbei.

Wir sind nun schon mit einigen Befehlen von REDCODE vertraut. Den Rest finden wir in Tabelle 2. Es sind insgesamt zehn Befehle, und wenn man unsere bisherigen Erklärungen erst einmal verdaut hat, dürfte sich die Bedeutung der restlichen Befehle fast von selbst erklären.

KNIRPS und GNOM stehen für eine Klasse von Programmen, die zwar nicht so wahnsinnig intelligent sind, dafür jedoch ausgesprochen aggressiv (so etwas soll es ja geben). Die nächste Klasse von Programmen könnte vielleicht so aussehen, daß sie zwar nicht ganz so aggressiv wären, dafür jedoch schlau genug, um unseren kleinen

TABELLE 2: REDCODE komplett

Anweisung	Kürzel	Argumente	Erklärung
Übertrage	MOV	A B	Übertrage Inhalt von Adresse A auf Adresse B
Addiere	ADD	A B	Addiere Inhalt von Adresse A zu Adresse B
Subtrahiere	SUB	A B	Subtrahiere Inhalt von Adresse A von dem von Adresse B
Springe	JMP	A	Übergib die Ausführung an Adresse A
Springe, wenn null	JMZ	A B	Übergib die Ausführung an Adresse A, falls Inhalt von Adresse B null ist
Springe, wenn größer	JMG	A B	Springe nach A, falls Inghalt von B größer als null ist.
Vermindere: springe, wenn null	DJZ	A B	Ziehe vom Inhalt der Adresse B 1 ab: Springe nach A, wenn B dann null ist.
Vergleiche	CMF	A B	Vergleiche Inhalt der Adressen A und B bei Ungleichheit überspringe nächste Anweisung
Data-Anweisung	DAT	B	Nichtausführbare Anweisung B ist der Datenwert

Tabelle 2: REDCODE komplett

Nervensägen, den KNIRPSEN und GNOMEN, auszuweichen, indem sie sich aus deren Reichweite herauskopieren.

Wie auch immer, alle Kampfprogramme unterliegen denselben Gesetzen: Sie bestehen aus einer Folge von REDCODE-Befehlen, die durch den MARS sequentiell abgearbeitet werden. Ein Befehl wiederum setzt sich aus einem Befehlskürzel (MOV, ADD etc.) sowie einem oder zwei Argumenten zusammen, denen ggf. ein Vorzeichen zur Bestimmung der Adressierungsart vorangeht. Dabei kann es sich um direkte, indirekte oder unmittelbare Adressierung handeln.

Da allgemein das Prinzip der relativen Adressierung Verwendung findet, hat kein Programm die Möglichkeit, seine Absolutposition in dem ringförmigen Speicher festzustellen. Es kann sozusagen niemals über den Rand seines eigenen Universums hinausblicken.

Dies soll als Einführung in den „Krieg der Kerne" genügen. Als Ergänzung können die beiden bereits erwähnten Artikel im „Spektrum der Wissenschaft" empfohlen werden.

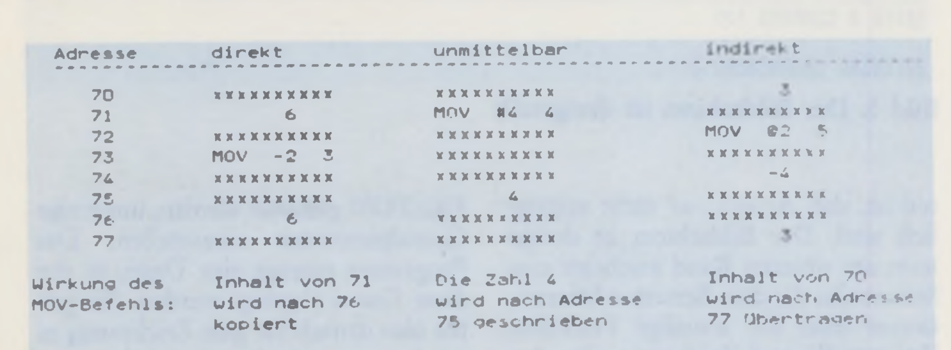
Das von uns erstellte Programm übernimmt die Übersetzung der mit einem Editor geschriebenen Programme, läßt sie gegeneinander kämpfen und zeigt das Ergebnis an. Es handelt sich um ein GEM-Programm; seine Drop-Down-Menüs ermöglichen die Einstellung einiger Parameter und Optionen. Auf der Public-Domain-Diskette befindet sich der „Krieg der Kerne" sowie die Bedienungsanleitung zum Programm.

Da wir zu zweit auf die Idee zum „Krieg der Kerne" gestoßen sind, haben wir auch die Programmierung gemeinsam durchgeführt. Dies war eine gute Erfahrung, und wir können es jedem ans Herz legen, einmal mit einem Partner ein Programmierprojekt durchzuführen. Man wird dadurch gezwungen, seinem Programm ein deutliches Konzept zu geben. Es erfordert auch erhebliche Disziplin, sich mit dem Partner über modularen Aufbau, Schnittstellen zwischen einzelnen Programmteilen usw. zu einigen, und es kommt nach unserer Erfahrung auch dem Programmierstil zugute. Sogenannter „Spaghetti-Code" rächt sich hier sehr schnell, so daß man besser von Anfang an übersichtlich programmiert.

Die relativ lange Programmierdauer von rund vier Monaten war nicht nur darin begründet, daß wir von Natur aus faul sind, sondern auch durch die Tatsache, daß die Programmierung von GEM-Programmen – trotz vieler Bücher und Kurse zu diesem Thema – für den gewöhnlichen Amateurprogrammierer noch immer ein ausgesprochenes Abenteuer darstellt. Will man neben den relativ einfach zu handhabenden VDI-Funktionen noch mehrere Windows mit ihren verschiedenen Möglichkeiten der Manipulation einsetzen, dann nimmt der erforderliche Mehraufwand in der Programmierphase – sowohl den Umfang als auch die Komplexität betreffend – schnell Formen an, die einen in einer schwachen Minute wehmutsvoll auf die guten alten BASIC-Zeiten zurückblicken lassen, als sich fast jeder Benutzerdialog durch die Befehle PRINT, GET und INPUT abwickeln ließ.

Trotzdem sollte niemand dieses Abenteuer scheuen. Mit etwas Mut und – wie gesagt – vielleicht mit einem Partner zusammen kann man es schon schaffen. Der Benutzer eines so erstellten Programms wird es später danken.

(Christian Hoofe / Ralf Hauke)



Adresse	direkt	unmittelbar	indirekt
70	xxxxxxxxxx	xxxxxxxxxx	3
71	6	MOV #4 4	xxxxxxxxxx
72	xxxxxxxxxx	xxxxxxxxxx	MOV @2 5
73	MOV -2 3	xxxxxxxxxx	xxxxxxxxxx
74	xxxxxxxxxx	xxxxxxxxxx	-4
75	xxxxxxxxxx	4	xxxxxxxxxx
76	6	xxxxxxxxxx	xxxxxxxxxx
77	xxxxxxxxxx	xxxxxxxxxx	3
Wirkung des MOV-Befehls:	Inhalt von 71 wird nach 76 kopiert	Die Zahl 4 wird nach Adresse 75 geschrieben	Inhalt von 70 wird nach Adresse 77 übertragen

Tabelle 3: Halb so schwer: Die Adressierungsarten

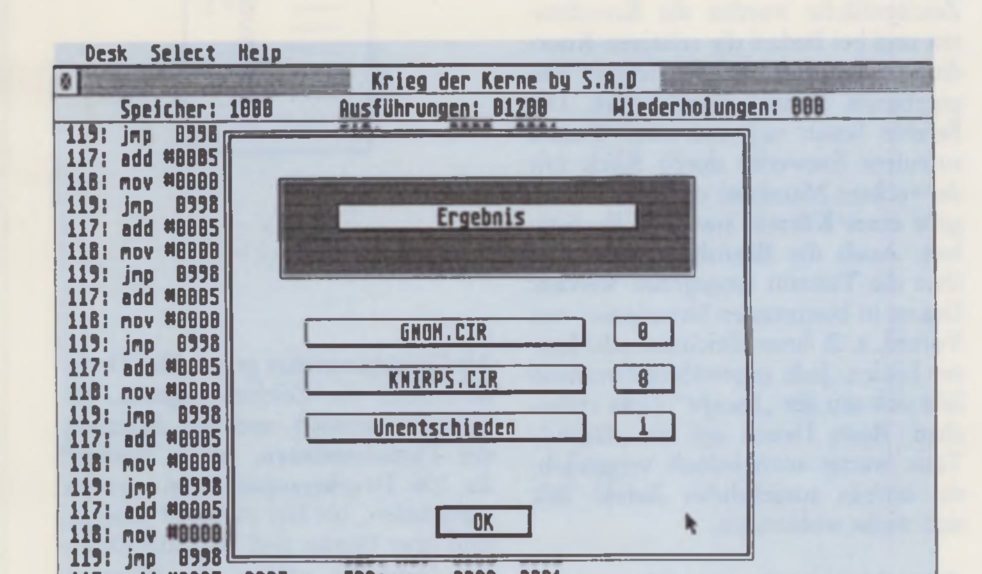


Bild 4: Ein Kampf ist zu Ende



First Cadd

CAD (Computer Aided Design) gewinnt auch im Bereich der Tischrechner immer mehr an Bedeutung. Während „richtige" CAD-Systeme mehrere hunderttausend DM kosten, muß man für PC-Lösungen „nur" einige zehntausend veranschlagen. Was leistet also ein CAD-Programm für den ST, das ca. 250 DM kostet? Wir testeten „First Cadd" von Generic Software.

Vorweg: Nimmt man das Kürzel CAD beim Wort, so hält First Cadd, was es verspricht, vergleicht man es aber mit den „großen Brüdern", die es beispielsweise für den IBM PC gibt, so sind schnell Grenzen erreicht. Das betrifft vor allem das Erstellen und Editieren von Zeichnungen.

Das Programm wird auf einer einseitigen Diskette geliefert, ein deutsches Handbuch gehört dazu. Letzteres ist ausreichend ausführlich, erklärt jedoch leider nicht alle Funktionen. So fehlt jeglicher Hinweis auf die Funktion „Cursor Size", mit der man die Cursorgröße bis zum bildschirmfüllenden Fadenkreuz verändern kann. Obwohl das Handbuch es bestreitet, lassen sich sechs verschiedene Linientypen darstellen (siehe Bild). Eine wertvolle Hilfe ist das umfangreiche Stichwortregister.

Da sich das Programm mit 512 kB Ram begnügt, ist es auch für die Besitzer der „kleinen" ST's interessant. Wer mehr Speicher zur Verfügung hat, kann es von einer Ram-Disk aus starten. Dieses empfiehlt sich besonders in der Einarbeitung, da einige Grundparameter und die Druckeranpassung in einem gesonderten Programm eingestellt werden.

Beim Programmstart fällt sofort ins Auge, daß es nicht unter GEM arbeitet – für den ST-Benutzer eine etwas ungewohnte Umgebung. Der Grund dafür liegt darin, daß es sich bei First Cadd um eine Adaption des für den IBM-PC erhältlichen „Generic CADD" handelt. Man gewöhnt sich jedoch schnell an die Menüführung. Der Nachteil ist, daß Accessories nicht zugänglich sind. Der Bildschirm ist dreigeteilt: am unteren Rand erscheint eine Statuszeile, die dem Benutzer Informationen über die jeweilige Funktion, Zoomgröße und Zeichnung gibt. Am rechten Rand befindet sich das Hauptmenü, hinter dessen einzelnen Punkten sich noch weitere verbergen. In die Zeichenfläche werden die Koordinaten und bei Bedarf die relativen Koordinaten (bezogen auf den letzten eingegebenen Punkt) eingeblendet. Die Befehle lassen sich auf zwei Weisen aufrufen: Entweder durch Klick mit der rechten Maustaste oder durch Eingabe eines Kürzels aus zwei Buchstaben. Auch die Koordinaten können über die Tastatur eingegeben werden. Das ist in bestimmten Situationen von Vorteil, z. B. beim Zeichnen sehr kurzer Linien. Jede angewählte Funktion läßt sich mit der „Escape"-Taste abbrechen. Beim Druck auf die „Undo"-Taste wartet man jedoch vergeblich: ein bereits ausgeführter Befehl läßt sich nicht widerrufen.

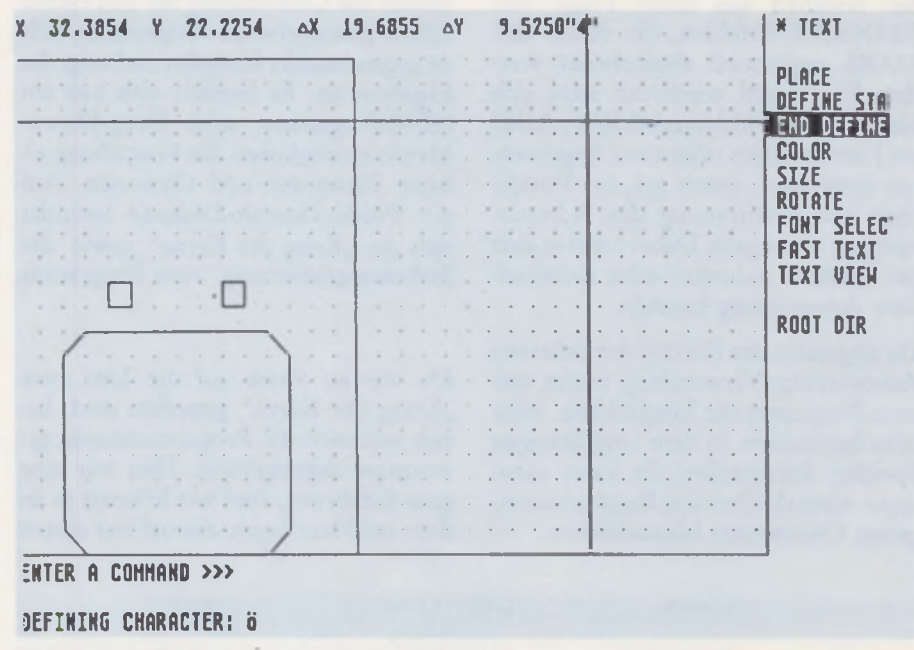
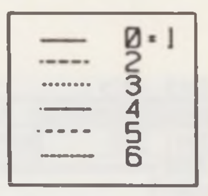


Bild 1: Der Bildschirm ist dreigeteilt

Grundparameter einstellen

Bevor die erste Zeichnung begonnen wird, sollte das eigenständige „CONFIG.TOS" gestartet werden, um einige Grundparameter einzustellen. Das Programm erzeugt eine Datei, in der diese Daten abgelegt werden. Sie gelten also danach für jede Zeichnung, es sei denn, man ändert sie wieder.



Als Grundparameter gelten: Der Druckertreiber, die Zeichnungsgröße, die Grundmaßeinheit und die Belegung der Funktionstasten. Nicht weniger als 106 Druckeranpassungen werden mitgeliefert, bis hin zum HP Laserjet sind über Plotter und 24-Nadel Drukker (Toshiba) nahezu alle gängigen Ausgabegeräte anzupassen. Da aber nur die wenigsten ST-Benutzer über Plotter oder ähnlich kostenintensive

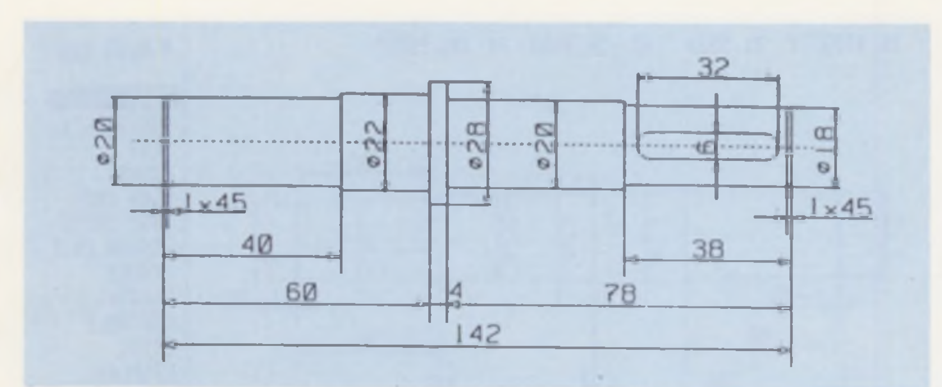


Bild 2: Sechs verschiedene Linientypen lassen sich darstellen

Druckgeräte verfügen, liegt das Schwergewicht bei den 9-Nadel Druckern. Für Epson und Kompatible gibt es sogar Treiber in verschiedenen Auflösungsstufen. Als Grundmaßeinheit sind Zoll und Millimeter vorgesehen. Leider rechnet das Programm offensichtlich intern in Zoll, so daß auf dem Weg zwischen Ein- und Ausgabe Ungenauigkeiten entstehen und es praktisch nicht möglich ist, exakte metrische Werte mit der Maus einzugeben. Die Funktionstasten lassen sich mit zehn beliebigen Befehlen belegen. Das sollte aber erst nach einiger Arbeit mit dem Programm getan werden, wenn man also die oft benötigten Befehle kennt.

Zeichnen

An Zeichenfunktionen bietet First Cadd die üblichen Punkte, Linien, Rechtecke, Kreise durch zwei und drei Punkte, Kreisbögen durch drei und vier Punkte sowie Ellipsen. Als Besonderheiten stellen sich regelmäßige Vielecke und Kurven durch eine beliebige Anzahl Punkte dar.

Gezeichnet wird auf bis zu 255 Ebenen (Layer), die immer genau übereinander liegen, deren Inhalt aber optional gezeigt wird. So kann eine Zeichnung, die aus mehreren anderen besteht, auf verschiedenen Ebenen erstellt werden. Zum Schluß werden alle gemeinsam dargestellt.

Durch reichhaltige Zoombefehle wie Darstellung der gesamten Zeichnung, der gesamten Zeichenfläche, Heranzoomen eines Fensters, u. a. ist immer der richtige Blick auf die Zeichnung gewährleistet. Es besteht aber auch die Möglichkeit, einen Maßstab manuell einzugeben.

Ebenfalls ist ein Raster (Grid) variabler Größe vorhanden, das Einrasten des Cursors darauf sowie die Darstellung des Rasters können ein- und ausgeschaltet werden.

Auch die Darstellung der Zeichnung während der Bearbeitung ist gut den jeweiligen Erfordernissen anzupassen. So ist es möglich, die Konstruktionspunkte (z. B. Mittelpunkte von Kreisen, Endpunkte von Linien, usw.) durch kleine Kreuze besonders darzustellen. Zum schnelleren Aufbau der Zeichnung kann der Text abgeschaltet werden.

Editieren

Die Edit-Funktionen sind ebenso reichhaltig ausgestattet: Es wird zwischen dem Editieren von Fenstern (die man mit Maus setzt) und dem von Elementen unterschieden. Als Element gelten die Grundbausteine einer Zeichnung, also Linien, Kreise, Bögen, Rechtecke, usw. Sowohl Fenster wie Elemente können kopiert, verschoben, verändert und gelöscht werden. Mit Hilfe der „Verändern"-Funktion ist es möglich, das Objekt auf eine andere Zeichenebene zu kopieren. Eine Besonderheit ist das Auftrennen von Linien. So kann man aus bereits bestehenden Linienzügen Teile heraustrennen, z. B. aus einem Kreisbogen ein Viertel „ausschneiden" oder Linien kürzen. Mit „Move Point" besteht die Möglichkeit, einen einzigen Punkt eines Elementes zu verschieben und so ein ganzes Objekt zu verzerren.

Eingebauter Font-Editor

Hinter dem Menüpunkt „Text" verbergen sich Befehle zum Verändern der Textgröße, des Winkels und zum Editieren von Textzeichen bzw. ganzen Zeichensätzen. Wer häufig mit kleinen Symbolen arbeitet, wie z. B. die Maßpfeile in der Beispielzeichnung, kann diese auf bestimmte Tasten legen. Der Größe der neuen „Textzeichen" sind aber keine Grenzen gesetzt.

Symbole

Eine der wichtigsten Funktionen der CAD-Programme ist das Definieren von Symbolen. Bei First Cadd heißen sie „Komponenten".

Als Symbol werden (Teil-) Zeichnungen definiert, die häufig in einer Zeichnung auftauchen, z. B. Schrauben, Kugellager, usw. Diese Symbole können einzeln abgespeichert und in jeder Zeichnung hinzugeladen werden. Dabei ist es möglich, sie in der Größe zu verändern, zu drehen und zu spiegeln.

Komfortables Drucken

Soll eine Zeichnung gedruckt werden, stehen mehrere Möglichkeiten zur Verfügung: Man kann die gesamte Zeichnung auf die Papiergröße angepasst, den momentanen Bildschirminhalt oder einen Ausschnitt in einem beliebigen Maßstab drucken. Es ist dann jeweils möglich, sich eine „Druckvorschau" geben zu lassen, um zu sehen, wie die Zeichnung auf dem Papier angeordnet sein wird.

Wurde im Konfigurationsprogramm der richtige Druckertreiber geladen, so steht dem Ausdruck nichts mehr im Wege. Mit dem Star NL10 und dem Epson-Treiber der höchsten Auflösung haben die Ergebnisse eine gute Qualität.

Menüs editierbar

Mit zwei besonderen Funktionen wartet First Cadd auf: Die Menüs sind frei editierbar, da ihre Zusammensetzung als ASCII-Datei abgespeichert wird. Es können also mit einem Editor eigene Menüs erstellt werden. Zum anderen kann man eine Zeichnung als ASCII-Text abspeichern und weiterverarbeiten.

Fazit

Wie bereits erwähnt, hält First Cadd, was es verspricht – zumindest insofern, als es den Benutzer beim Erstellen einer Zeichnung unterstützt. Beim Zeichnen selber fällt auf, daß Funktionen, die häufig gebraucht werden, feh-

len. Dazu gehören insbesondere Tangenten an Kreise, Lote auf Kreise und auf Linien, sowie Mittelsenkrechten. Ferner fehlt eine Trimmfunktion, d. h., daß Linien, die sich zwar kreuzen, aber evtl. etwas zu lang geraten sind, automatisch gekürzt werden, so daß sich eine genaue Ecke ergibt. Wie in der Beispielzeichnung zu sehen, besteht ein großer Teil der Arbeit an technischen Zeichnungen aus dem Bemaßen. Zwar verfügt First Cadd über eine Funktion, mit der man Längen, Winkel und Flächen messen kann, es bietet jedoch keine automatische Erzeugung von Maßlinien oder wenigstens Pfeillinien, von einer automatischen Bemaßung ganz zu schweigen. So wird das Bemaßen zu einer zeitraubenden Arbeit, bei der man die Pfeile an den Maßlinien improvisieren muß.

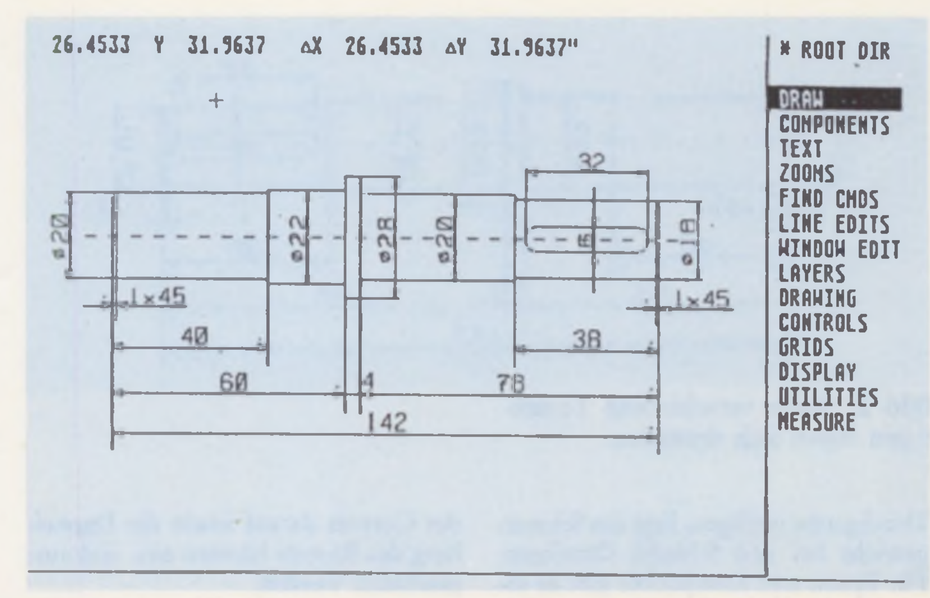


Bild 3: Solche Zeichnungen kann man auch mit 9 Nadeln drucken

Das größte Manko jedoch ist die fehlende Schraffurfunktion. Wer jemals eine technische Zeichnung gesehen hat, dem wird aufgefallen sein, daß jede Fläche, die durch einen Schnitt durch ein Bauteil zustande kommt, schraffiert dargestellt wird. Ohne eine Schraffur kann man also nur einen kleinen Teil seiner Zeichnungen wirklich normgerecht erstellen.

Abgesehen von diesen Fehlern, wird dem Benutzer ein durchdachtes Instrument an die Hand gegeben, mit dem er bereits nach kurzer Einarbeitung schnell und sauber arbeiten kann. Als besonders praktisch haben sich dabei erwiesen: die Möglichkeiten, Befehle über die Tastatur einzugeben, die vielen Zoomfunktionen, das Herausschneiden von Linienteilen sowie die Druckvorschau. Leider zeigte sich das Programm bei Fehlbedienungen nicht ganz absturzsicher.

Aufgrund fehlender, wirklich wichtiger Funktionen muß man aber sagen, daß First Cadd kein ganz komplettes Werkzeug ist. Es ist zu hoffen, daß bald eine verbesserte Version erscheint, die die noch fehlenden Funktionen mit einschließt.

Ingo Brümmer

First Cadd

Entwickler: Generic Software Inc.

Vertrieb: PublicSoft

Preis: ca. 250 DM

DELA Elektronik

Drucker

Star NL 10 inkl. Interface	DM	549,–
Citizen 120 D	DM	199,–
Seikosha SL-90AI	DM	798,–
DELA-Printer Parallel	DM	549,–
NEC P6	DM	1.049,–

Monitore

NEC Multisync	DM	1.298,–
Philips 8833	DM	678,–

Disketten (100er Pack)

5"25 MD 1D	DM	75,–
5"25 MD 2D	DM	85,–
3"5 2DD	DM	250,–

In unseren Filialen zahlen Sie entsprechende Preise auch für den 10er Pack!!!

Diskettenlaufwerke-Festplatten

Diskettenlaufwerk f. Atari ST	DM	399,–
Amiga-Diskettenlaufwerk 3"5 Disk	DM	369,–
Seagate ST225 inkl. Controller 20 MB	DM	666,–
Seagate ST238 inkl. Controller 30 MB	DM	749,–
Seagate ST251 40 MB 40 ms	DM	838,–

Modems & Akustikkoppler & BTX

(Modems o. FTZ-Nummer)

IBM-Modem 1200/75 Bd	DM	199,–
Dataphon S 21-23d	DM	298,–
BTX-Term f. IBM	DM	198,–

Wir führen Commodore-Chips, Eproms, elektronische Bauteile u. v. m.
Fordern Sie unsere Preisliste an.
Außerdem führen wir reichhaltiges Zubehör für Commodore-, Atari-, Schneider- und IBM-Computer – Info unbedingt anfordern!

Sound Sampler für Amiga

Neben einer professionell gefertigten Hardware erhält das Komplettpaket eine komfortable Software.
Abspeichern der Daten als Objektfile.
Generierung von Sound-Disketten.
(Option: mit 2 Laufwerken kann unbegrenzt lange und ohne Unterbrechung gesampelt werden.)
Komprimiermöglichkeit (spart bis zu 50% Speicherplatz).
Erstellung von IFF-Files möglich; lädt jedes File (nicht nur IFF-Format).
Programm erkennt und unterstützt auch Speichererweiterungen.
Weitere Optionen: das Mixen der Sounds.

Das Alles naürlich zum DELA-Preis von DM **89,–**

Sound Sampler Software

Für alle, die bereits die entsprechende Hardware besitzen.

DELA-Preis DM **49,50**

DELA-AMIGA 500-EPROMMER

In Kürze bei DELA erhältlich:
der Eprommer für den AMIGA 500.
Natürlich mit den gleichen überragenden Leistungsmerkmalen wie der DELA-ATARI-ST EPROMMER.

Natürlich zum DELA-Preis: DM **149,–**

DELA-Atari ST Epromdisk

Vereinigt die Geschwindigkeit einer Ramdisk mit der Zuverlässigkeit von ROM's.
Voll Software-gesteuert durch mitgeliefertes Treiberprogramm und Modulgenerator.
Auch für AUTO-Ordner und ACC-Dateien.
Die Grundversion von 512 K-Byte kann durch ein Aboveboard auf 1 M-Byte erhöht werden.
DIE GESCHWINDIGKEIT DER EPROMDISK IST HÖHER ALS DIE EINER FESTPLATTE!!!
Natürlich mit ausführlicher Anleitung für ein kinderleichtes Handling und zum

DELA-Preis DM **99,–**

DELA-Shugate-Interface für Amiga

Jetzt wird der Anschluß von 3"5 und 5"25 Laufwerken leichtgemacht.
Einfach mit Laufwerk und Computer verbinden – fertig!!!
Auch für Doppellaufwerke geeignet!

DELA-Preis DM **49,50**

NEU AB 1. DEZ. 1987
DER DELA-MASTER-CLUB
Fordern Sie Informationen an.



DELA-ATARI-ST EPROMMER

Technische Daten:

- Programmierspannung: 21 V und 12.5 V Softswitch
- kein Netzteil erforderlich
- vollkommen Software-gesteuert keine unprofessionellen Schalter
- echter hochwertiger TEXTOOL Sockel
- eingebauter Generator für Atari-Rom-Module
- Epromtypen: 2764, 27128, 27256, 27512, 27513, 27011 natürlich auch alle kompatiblen Typen und die entsprechenden CMOS, sowie A-, B-, C-Typen
- Normal-Mode, Schnell-Mode, Einzelbyte-Mode ...
- wird einfach in den Romport gesteckt, der Druckerport bleibt natürlich frei

und die absolute Sensation:
der vorläufige Preis für dieses SUPERDING: DM **149,–**
(in Worten: eins-vier-neun DM!!!)

DELA-Atari ST-Userport

Die Schnittstelle zur Außenwelt (z.B. für Steuerungsaufgaben).
Leistungsmerkmale:

- 2 Steckleisten (insgesamt 64 bit I/O)
- je Steckerleiste 2x8 bit Input und 2x8 bit Output
- TTL-kompatibel und gebuffert
- In allen Programmiersprachen frei programmierbar!!!
- Ausführliche Anleitung mit vielen Beispielen

DELA-Preis DM **99,–**

Besuchen Sie uns mal in:

Köln 1 · Maastrichter Straße 23
Essen 1 · Schützenbahn 11–13 DGB-Haus Porscheplatz
München 22 · Bürkleinstraße 10

Fordern Sie unseren Katalog an – für alle Informationen – mit allen Preisen.

DELA immer aktuell, immer preiswert:

Am besten gleich bestellen!



24 Stunden Anrufbeantworter 02 21 - 7 15 17 30
Mailbox: 02 21 - 7 15 17 40
Telefon: 02 21 - 7 15 17 / 20-22
Telefax: 02 21 - 7 15 17 60
Teletex: 2 214 248 DELA

Versandbedingungen:
Versand ab 40,– DM, sonst nur gegen Vorkasse.
Nachnahmeversand NN-Spesen 7,50 DM, b. Vorkasse 4,00 DM.
Auslandsbestellungen nur gegen Vorkasse + 15,00 DM Versandkosten. Wir liefern auf Ihre Rechnung und Gefahr zu den Verkaufs- und Lieferbedingungen des Elektronikgewerbes.
Preis: Stand 15.10.1987

Bestellcoupon ausfüllen, auf Postkarte kleben oder im Umschlag an:
DELA Elektronik · Merkenicher Straße 87–89
5000 Köln 60

Bestellcoupon

ST-Computer / 11/87

Menge	Artikel	Einzelpreis

Name ____ Anschrift ____

Kunden-Nr. ____ Datum ____ Unterschrift ____

Einführung in Forth

Teil 6

Die Serie begann mit der Beschreibung einer Ampelsteuerung. In der letzten Folge der Serie soll diese Ampelsteuerung in die Praxis umgesetzt werden, zumal es ein Bereich ist, in dem der Haupteinsatzbereich von FORTH liegt, der industriellen Geräte- und Prozeßsteuerung bzw. -überwachung. Dabei lernen Sie auch die typische Vorgehensweise zur Entwicklung eines FORTH Programms kennen.

Um die Brauchbarkeit einer Programmiersprache bewerten zu können, empfiehlt es sich, die Hauptanwendungsgebiete der Sprache zu untersuchen. FORTH wurde ursprünglich zur Prozeßsteuerung konzipiert. Die Steuerung und Überwachung (industrieller) Anlagen und Geräte ist auch heute noch der Hauptanwendungsbereich von FORTH, obwohl FORTH andererseits keine typische Prozeßsteuerungssprache ist, denn es lassen sich genauso gut Textverarbeitungsprogramme oder Expertensysteme in FORTH implementieren. In dieser Folge soll es darum gehen, die Stärken von FORTH im Bereich der Gerätesteuerung zu demonstrieren. Zwar stehen als „Gerät" lediglich ein paar Leuchtdioden zur Verfügung, dennoch lassen sich bereits an einer simplen Ampelsteuerung die spezifischen Vorteile von FORTH erkennen. Vor allem soll deutlich werden, warum FORTH Sprachen wie BASIC, C oder ASSEMBLER in bestimmten Punkten überlegen ist. Nicht zuletzt lernen Sie an diesem anschaulichen Beispiel die einzelnen Phasen bei der Entwicklung eines FORTH Programms kennen. Eingefleischte ST COMPUTER Leser werden sich vielleicht noch daran erinnern, wie diese Einführungsserie begann. In der ersten Folge wurde der typische Aufbau eines FORTH Programms am Beispiel einer Ampelsteuerung veranschaulicht.

Es wäre sicher wenig sinnvoll gewesen, Ihnen bereits damals ein komplettes Listing für eine Ampelsteuerung in FORTH zu präsentieren. Mittlerer-weile, nachdem fünf Folgen vergangen sind, und Sie hoffentlich einiges über FORTH hinzugelernt haben, ist ein preiswerter Userport für den ST erhältlich, und einer praktischen Umsetzung des Programms steht eigentlich nichts mehr im Wege.

TOP DOWN Strategie

Wenn man am Anfang der Lösung eines größeren Problems steht, empfiehlt es sich, sich zunächst ein paar Gedanken über die grundsätzliche Vorgehensweise zu machen. Dies gilt für Situationen aus dem täglichen Leben, wie etwa das Vorbereiten auf eine Prüfung, insbesondere aber für das Erstellen eines Anwenderprogramms. Über die mögliche Vorgehensweise bei der Programmentwicklung sind bereits ganze Bücherschränke an Literatur verfügbar, in denen Dutzende verschiedener Strategien in allen Einzelheiten diskutiert werden. Als Quintessenz läßt sich zusammenfassen, daß es eine universell anwendbare Lösungsstrategie nicht gibt. Dies ist auch nicht verwunderlich, denn dazu sind die Probleme, die gelöst werden müssen, die Werkzeuge, mit denen diese Probleme gelöst werden sollen (in erster Linie die Programmiersprachen) und nicht zuletzt auch diejenigen Personen, die die Probleme lösen sollen, einfach zu verschieden.

Es gibt dennoch eine Vorgehensweise, deren Anwendung sich in den meisten Fällen als vorteilhaft erweist. Nach dieser Strategie wird zunächst einmal das Problem grob umrissen und die einzelnen Schritte des Programmablaufs werden namentlich festgelegt. Zweckmäßigerweise wird das gesamte Programm in einzelne Module aufgeteilt, wobei jeder der ermittelten Schritte mit einem Modul identisch ist. Die Aufteilung in Module bringt u. a. folgende Vorteile :

- das Programm wird überschaubarer
- die Fehlersuche wird erleichtert, da Module einzeln getestet werden können
- durch einfaches Austauschen einzelner Module kann das Programm jederzeit erweitert bzw. verändert werden

In der Planungsphase werden die Module noch als „Schwarze Kästen" behandelt, bei denen lediglich die Funktion und unter Umständen noch die Ein-Ausgabeparameter interessieren. Im nächsten Schritt wird der Programmablaufplan weiter verfeinert, indem jedes Modul in weitere, in der Regel kleinere, Module unterteilt wird. Dieser „iterative Zyklus" wiederholt sich solange, bis eine weitere Verfeinerung nicht mehr möglich ist bzw. nicht mehr sinnvoll erscheint. Nun, und nicht eher, sollte mit der Codierung d.h. dem Umsetzen des aufgestellten Programmablaus in die Befehle und Operationen einer beliebigen Programmiersprache begonnen werden.

Die eben beschriebene Entwurfsstrategie wird als TOP-DOWN Entwurf bezeichnet, womit die Vorgehensweise, nämlich die wiederholte Verfeinerung eines allgemeinen Lösungsansatzes, recht treffend beschrieben wird. Dabei ist der TOP-DOWN Entwurf kein einmaliger Vorgang. Er wiederholt sich quasi auf jeder Ebene erneut, d. h. „Untermodule" werden nach dem gleichen Muster in weitere Untermodule zerlegt usw.. Der Planunsphase schließt sich die Codierungsphase an, in der das eigentliche Programm geschrieben wird. Allgemein gilt: daß je besser und strukturierter der Programmablauf entworfen wurde, desto leichter die Umsetzung in ein konkretes Programm.

Um die ganzen Betrachtungen nicht allzu theoretisch erscheinen zu lassen, soll gleich mit dem eingangs angekündigten Beispiel begonnen werden. Stellen Sie sich vor, daß Sie als Leiter einer Softwareentwicklungsabteilung für einen Kunden eine Steuerung für eine komplette Ampelanlage einer typischen Verkehrskreuzung entwerfen sollen. Sie beginnen gemäß dem vorhin beschriebenen TOP-DOWN Entwurf zunächst mit einer allgemeinen Beschreibung des Problems (Festlegen der be-

nötigten Ampeln und deren Phasen) und gelangen nach mehreren Entwurfszyklen zu dem Problem, das Verhalten einer einzelnen Ampel beschreiben zu müssen. Dies könnte am einfachsten nach folgendem Schema erfolgen:

```
: AMPEL
  WIEDERHOLE
    GRÜN__PHASE
    GELB__PHASE
    ROT__PHASE
    ROT/GELB__PHASE
  ENDE-WIEDERHOLE
;
```

Damit wäre der Ablauf beschrieben und der Entwurf auf der obersten Ebene komplett. Gleichzeitig haben wir aber schon ein, wenigsten im Prinzip, ablauffähiges FORTH Programm vorliegen. Daraus läßt sich eine wichtige Erkenntnis ableiten :FORTH verkürzt die Planungsphase eines Programms erheblich, da sich Planungsphase und Codierungsphase nicht mehr ausschließen, sondern sich teilweise überlappen.

Nun wiederholt sich der Entwurfszyklus, allerdings eine Ebene tiefer. Konkret heißt das, daß wir nun alle Komponentenworte des Wortes AMPEL verfeinern müssen. Konkret könnte das folgendermaßen aussehen:

```
: GRÜN__PHASE
    GRÜNE__LAMPE EIN
    WARTEN
    GRÜNE__LAMPE AUS
;
```

Auch hier wurde ein Ablauf in einer Art und Weise entworfen, die auf der einen Seite der menschlichen Sprache sehr ähnlich ist und auf der anderen Seite direkt in FORTH codiert werden kann. Dadurch wird die Entwicklunsphase eines FORTH Programms erheblich beschleunigt. Hinzu kommt, daß mit FORTH als interaktive Entwicklungsumgebung nicht der klassische und zeitaufwendige „Edit-Compile -Link -Go -Zyklus" wie bei Sprachen wie C oder MODULA-2 durchlaufen werden muß.

Ähnlich wie das Wort GRÜN__PHASE werden auch die übrigen Worte GELB__PHASE, ROT__PHASE und ROT/GELB__PHASE definiert. Als kleinen Nebeneffekt sollten Sie folgende Regel beherzigen, die zwar auf den ersten Blick als ein wenig trivial erscheinen mag, die aber die Lesbarkeit und das

```
Screen #1

 0  Loadscreen                                  *  20/08/87  PM
 1
 2  2  7  THRU
 3
 4
 5  \ dies sieht zwar nach Platzverschwendung aus
 6
 7  \ hält aber die Möglichkeit späterer Erweiterungen offen
 8
 9
10
11
12
13
14
15

Screen #2

 0  Ampelsteuerung - Konstantendefinition *  20/08/87  PM
 1
 2  $FF8604  CONSTANT  DMA_DATA
 3  $FF8606  CONSTANT  DMA_MODE
 4  $43E     CONSTANT  FLOCK
 5
 6  1 CONSTANT  ROTE_LAMPE
 7  2 CONSTANT  GELBE_LAMPE
 8  4 CONSTANT  GRÜNE_LAMPE
 9
10  VARIABLE USER_PORT
11  VARIABLE GELB_FLAG
12
13
14
15

Screen #3

 0  Ampelsteuerung - Initialis. u. Warteschl.   *  20/08/87  PM
 1
 2 : DMA_INIT ( --- )
 3    $FF FLOCK W!           \ Floppyzugriff sperren

 4    $188 DMA_MODE W!       \ DMA Port konfigurieren
 5 ;
 6
 7 : TU_NIX ( --- )
 8 ;                         \ der Name sagt's bereits
 9
10 : WARTE  ( n --- )
11    DO TU_NIX LOOP
12 ;
13
14
15

Screen #4

 0 Ampelsteuerung  - Userport Ansteuerung  * 20/08/87
 1
 2 : EIN  ( c --- )
 3    USER_PORT C@              \ Zustand des Userports
 4    OR                        \ mit Muster verknüpfen
 5    DUP                       \ Wert kopieren
 6    USER_PORT C!              \ neuen Zustand speichern
 7    DMA_DATA W! ;             \ auf Userport ausgeben
 8
```

Abbildung 1: Das Ampelsteuerungs Listing

Verständnis von FORTH Code langfristig enorm steigern kann: Verwenden Sie Wortnamen mit Bedacht und nutzen Sie den Umstand, daß Wortnamen bis zu 31 Zeichen enthalten können aus. Schreiben Sie ruhig z. B. „GRÜN_PHASE_EINSCHALTEN" (22 Zeichen) anstelle von z. B. „GP_EIN".

Die nächste Phase der Verfeinerung wird bereits in irgendeiner Form die Hardware, d. h. in unserem Beispiel den ST USERPORT ansprechen. Spätestens hier beginnt die Phase der Codierung, d. h. ein allgemeiner Entwurf ist in diesem Stadium nicht mehr sinnvoll. Anders als der Entwurf verläuft die Implementierung in FORTH von unten nach oben. Der Name für diese Strategie ist BOTTOM-UP und soll den Umstand beschreiben, daß mit der Implementation des elementarsten Wortes der Applikation begonnen wird. Daraufhin folgen die Worte, die auf diesen elementaren Worten aufbauen, usw. Die Implementierung endet in FORTH mit einem einzigen Wort, welches bei seinem Aufruf das gesamte Anwenderprogramm zur Ausführung bringt.

Das komplette Listing finden Sie in Abbildung 1. Zunächst noch einmal eine kurze Beschreibung des DMA Ports des ATARI ST, für den Fall, daß Sie das April Heft nicht vorliegen haben. Der DMA Controller des ATARI ST belegt den Adreßbereich von $FF8600 bis $FF860D. Für die Programmierung des Userports sind nur die beiden Register $FF8604 und $FF8606 interessant. Bei ersterem handelt es sich um das Datenregister, dessen Inhalt den Zustand der einzelnen Pins des Userports festlegt; bei dem letzterem um das DMA Kontrollregister, welches u. a. festlegt, ob eine Eingabe oder eine Ausgabe erfolgt. Um nun ein bestimmtes Bitmuster auf dem Userport ausgeben zu können, muß über das Kontrollregister der DMA Port auf Ausgabe geschaltet und das entsprechende Bitmuster in das Register $FF8604 geschrieben werden. Um Komplikationen zu vermeiden, sollte ferner die Systemvariable FLOCK unter der Adresse $43E für die Dauer des Zugriffes mit $FF belegt werden. Damit wären die wichtigsten Voraussetzungen geklärt und der Umsetzung des Programms steht eigentlich nichts mehr im Wege.

```
 9 : AUS  ( c --- )
10     USER_PORT C@                    \ Zustand des Userports
11     255 SWAP - AND                  \ mit Muster verknüpfen
12     DUP                             \ Wert kopieren
13     USER_PORT C!                    \ neuen Wert speichern
14     DMA_DATA W! ;                   \ auf Userport ausgeben
15


Screen #5

 0 Ampelsteuerung  - Ampelphasen     * 20/08/87
 1
 2 : GRÜN_PHASE ( --- )
 3   GRÜNE_LAMPE  EIN
 4   20000 WARTEN
 5   GRÜNE_LAMPE  AUS ;
 6
 7 : GELB_PHASE ( --- )
 8   GELBE_LAMPE EIN
 9   10000 WARTEN
10   GELBE_LAMPE AUS ;
11
12 : ROT_PHASE ( --- )
13   ROTE_LAMPE EIN
14   20000 WARTEN
15   ROTE_LAMPE AUS ;


Screen #6

 0 Ampelsteuerung - Ampelphasen    * 20/08/87
 1
 2 : ROT/GELB_PHASE
 3   GELBE_LAMPE EIN
 4   ROTE_LAMPE EIN
 5   10000 WARTEN
 6   GELBE LAMPE AUS
 7   ROTE_LAMPE AUS  ;
 8 : GELB_BLINKEN   ( --- )
 9   GELB_FLAG @ IF                   \ Gelbe Lampe an
10               0 GELB_FLAG !       \ Ja, dann Flag rücksetzen
11               GELBE_LAMPE AUS     \ und gelbe Lampe aus
12             ELSE                  \ ansonsten
13               1 GELB_FLAG !       \ Nein, dann Flag setzen
14               GELBE_LAMPE EIN     \ und gelbe Lampe ein
15             THEN  ;


Screen #7

 0 Ampelsteuerung  - Hauptwort    * 20/08/87  PM
 1
 2 : AMPEL  ( --- )
 3   DMA_INIT  ALLES_AUS  1 GELB_FLAG !
 4   BEGIN                \ Beginn der Endlosschleife
 5    ROT_PHASE
 6    ROT/GELB_PHASE
 7    GRÜN_PHASE
 8    GELB_PHASE
 9    KEY IF              \ Prüfen auf Tastendruck
10         BEGIN          \ Ja, dann Gelbblinken 11 GELB_BLINKEN KEY
12         UNTIL          \ Aufhören, wenn Taste gedrückt
13         THEN
14    AGAIN               \ Ende der Endlosschleife
15
```

Listing 1

Das Wort DMA_INIT muß am Anfang aufgerufen werden und nimmt die notwendige Initialisierung vor. Dieses Wort sowie die Worte EIN und AUS sind die eigentlichen hardwareabhängigen Worte. Sollten sich aus irgendeinem Grund die Hardwarespezifikationen ändern, so müssen lediglich diese drei Worte entsprechend angepaßt werden, unabhängig davon, ob Sie lediglich eine einzelne Ampel steuern oder eine ganze Kreuzungsanlage. Zu dem Wort EIN muß nicht viel gesagt werden, es testet zunächst anhand eines Flags. Ausgehend von einer bestimmten Anordnung des Userports (siehe Abb. 2) wird ein entsprechendes Bitmuster auf dem Userport erzeugt. Entsprechend arbeitet das Wort AUS. Diese Bitmuster ergeben sich aus den Codes für die einzelnen Phasen. Daher wurden diese Codes als Konstanten definiert, da man nun schreiben kann:

GRÜNE_LAMPE EIN OK

um z. B. die grüne Lampe einzuschalten. Nicht nur, daß diese Schreibweise eigentlich sehr logisch ist und auch von jemandem, der von FORTH keine Ahnung hat, angewendet werden kann, ist sie auch universell erweiterbar. Sollen nämlich irgendwann einmal mehrere Ampeln gleichzeitig gesteuert werden, so könnte die dazugehörige Syntax z. B. folgendermaßen aussehen:

ERSTE GRÜNE_LAMPE EIN
ZWEITE GRÜNE LAMPE EIN usw.

Bei ERSTE und ZWEITE handelt es sich um Konstanten mit den Werten eins und zwei. Das Wort GRÜNE_LAMPE hat zwar noch den gleichen Namen aber eine vollkommen neue Struktur. Es wertet den konstanten Wert, der durch ERSTE, ZWEITE usw. übergeben wird aus und verknüpft ihn so, daß ein entsprechendes Bitmuster erzeugt wird. Das Wort EIN ist nach wie vor unverändert. Ihm ist es gleichgültig, wie das Bitmuster zur Ansteuerung des Userports erzeugt wurde. Dieses „Verstecken von Information" (das Wort EIN „weiß" nicht, wie das Wort GRÜNE_LAMPE das Bitmuster erzeugt) und die extreme Modulasierung findet man auch in modernen allerseits hochgelobten Sprachen, wie z. B. MODULA-2. Bereits an diesem einfachen Beispiel wird die Mächtigkeit von FORTH deutlich, die unterschiedlichsten Datenstrukturen miteinander kombinieren zu können. So konnte das Wort GRÜNE_LAMPE, im ersten Fall noch eine Konstante, ohne weiteres auch als Befehl geschrieben werden, ohne daß dadurch der Aufbau des Programms geändert werden mußte. Diese Flexibilität bei der Implementation eines Programmentwurfs bzw. diese Anpassungsfähigkeit des Programmcodes an neue Situationen ermöglicht keine andere Sprache in diesem Umfang.

Loadscreens steuern den Programmfluß

Doch zurück auf den Boden der Realität bzw. der Praxis. Geben Sie das Listing mit Hilfe des Editors ein. Noch eine Anmerkung zu Screen 1: Hierbei handelt es sich um einen sog. „Loadscreen", der lediglich eine Reihe von LOAD Befehlen (in unserem Beispiel nur einen einzigen, der alle sechs Screens auf einmal lädt) enthält. Beim Laden des Loadscreens durch '1 LOAD' werden die einzelnen Screens in der Reihenfolge des Auftretens der korrespondierenden LOAD Befehle geladen. Der große Vorteil eines Loadscreens, besonders in umfangreicheren Applikationen, ist, daß man bei der Eingabe des Quelltextes nicht an eine starre Reihenfolge gebunden ist. Vielmehr kann der Quelltext in einer beliebigen Anordnung eingegeben werden. Die Reihenfolge der Kompilation wird lediglich durch den Loadscreen festgelegt. Zusätzlich können einzelne Screens einfach dadurch vom Kompilieren ausgeschlossen werden, daß ihr entsprechender LOAD Befehl aus dem Loadscreen entfernt wird.

Kein Kommentar ?

Die einzelnen Screens wurden weitgehend mit Kommentaren versehen. Auch dies ist etwas, was man sich als FORTH Programmierer nicht früh genug angewöhnen kann. Geizen Sie nicht mit Kommentaren, aus denen der Sinn einzelner Worte hervorgeht. FORTH bietet seinem Benutzer eine sehr große Freiheit. Doch darin liegt auch die Gefahr, sehr schnell unleserlichen Code entstehen zu lassen. Zeile 0 eines Screens sollte generell als Kommentarzeile benutzt werden, aus der der Inhalt des jeweiligen Screens hervorgeht.

Starten des Programms

Wenn Sie nun das Programm starten, so durchläuft die Ampel, die aus drei an den Userport angeschlossenen Leuchtdioden (oder auch Lampen mit entsprechenden Transistortreibern mit oder ohne Optokopplern, zur Trennung des Starkstromkreises voran) besteht, alle Phasen. Bei Betätigen einer Taste geht unsere Ampel in eine Gelbblinkphase über. Das Programm wurde der Einfachheit halber als eine Endlosschleife konzipiert, die Sie nur durch einen Reset verlassen können. Eventuell kann es erforderlich sein, die Warteschleifen der einzelnen Phasen zu verändern.

Eine neue Sprache entsteht

Durch die Worte aus den Screens 2 bis 7 wurde aus der Vielzwecksprache FORTH eine „Spezialsprache" zur Steuerung von Verkehrsampeln. Dieses Konzept geht weit über das Bibliothekskonzept anderer Sprachen hinaus, da die neuen Worte genauso Bestandteil der Sprache sind, wie die Kernworte und von diesen nicht unterscheidbar sind. Dem Anwender steht es nun frei, den erweiterten Sprachkern als eine sog. „Turnkey" Applikation zu speichern, die dafür sorgt, daß beim erneuten booten des Systems das Anwenderprogramm gestartet wird. Professionelle Anwender können mit Hilfe eines Metacompilers dedizierte Systeme kompilieren, die nur die Worte (dabei wird von diesen Worten der Wortkopf entfernt) enthalten, die für die Applikation benötigt werden. Als Ergebnis eines solchen Compilationsprozesses kann auch reiner Maschinencode entstehen. Durch Verwendung zusätzlicher „Targets" kann es sich hierbei um lauffähigen Maschinencode für im Grunde jeden beliebigen Prozessor, sei es für einen 8031 oder eine 68020, handeln.

Ende und Ausblick

Damit wäre die FORTH Einführungsserie vorläufig beendet. Ich möchte mich vor allem bei den Lesern bedanken, die bis zum Ende durchgehalten haben. Ich hoffe, es hat Ihnen Spaß gemacht, und Sie haben ein Gefühl dafür bekommen, was FORTH kann und was nicht. Sicher ist FORTH nicht „die" ideale Sprache. Doch, obwohl es

immer Anwendungen geben wird, die man besser in Assembler, BASIC, C, MODULA-2, PEARL oder LISP programmiert, kommt FORTH meiner Meinung nach einer solchen fiktiven Idealsprache sehr nahe. Im Grunde enthält es Komponenten aller aufgezählten Sprachen, bzw. kann jederzeit um diese Komponenten erweitert werden. Ich hoffe, Ihnen auch in Zukunft FORTH Programmiertechniken in Form praktischer Anwendungen präsentieren zu können. Wenn Sie Informationen über FORTH, Public Domain Software oder weiterführende Literatur benötigen, so wenden Sie sich an die

FORTH Gesellschaft e.V.
Friedensallee 92
2000 Hamburg 50

Schließen soll diese Einführungsserie mit einem Zitat, das mir persönlich sehr gut gefällt. Es stammt von Michael Ham, ehemaliger Kolumnist für Dr. Dobbs Journal, FORTH Enthusiast und Gewinner eines Wettbewerbs zur Beschreibung von FORTH in fünfundzwanzig oder weniger Worten. Ich habe es dem sehr empfehlenswerten Buch „In FORTH denken" von Leo Brody (erschienen im Hanser Verlag, 1986) entnommen:

„FORTH ist wie das Tao: ein Weg, den man erkennt, wenn man ihm folgt. Seine Fragilität ist seine Stärke, seine Einfachheit ist seine Botschaft."

(PM)

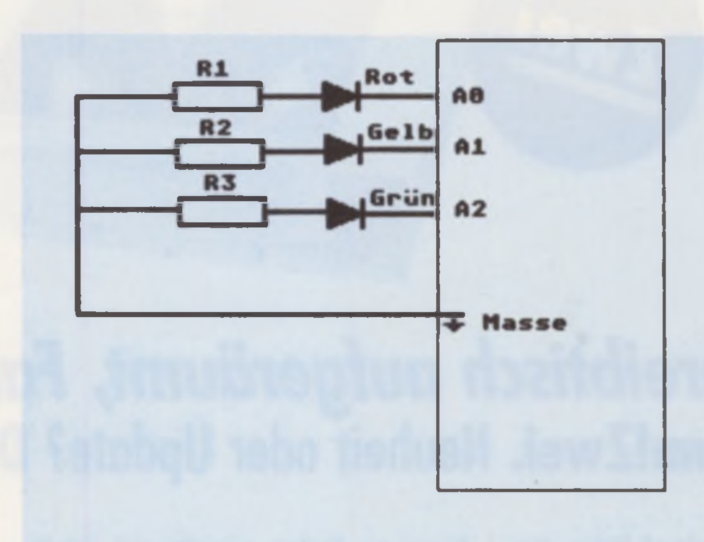
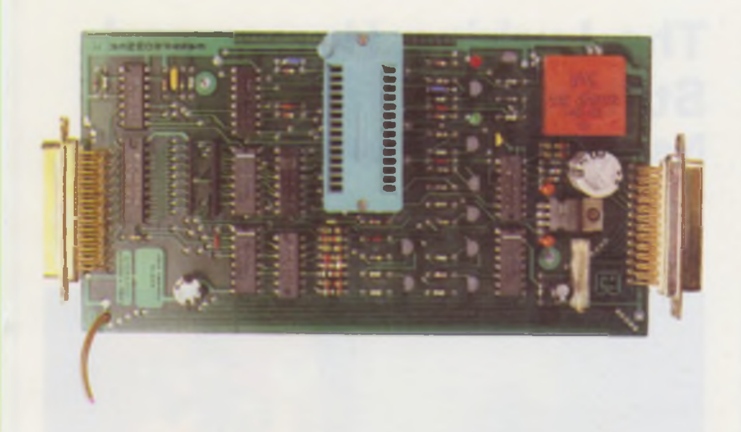


Abb.2 Belegung des Userports. (R1,R2,R3 = 470 Ohm)

Aus der Hardwareküche der





Die 576 KByte Easybank

Diese 'gebankte' EPROM-Karte erlaubt es den ROM-Speicher des ATARI ST um über **570 Kilobyte** zu erweitern. Damit haben Sie die Möglichkeit ganze Programmpakete sofort nach dem Einschalten des Rechners parat zu haben. Selbst das Booten von Autostart-Programmen und Accessories ist möglich. Auf der Easybank finden bis zu acht EPROMs des Typs 27512 oder 27256 Platz, wobei die Steuersoftware automatisch in zwei weitere EPROMs des Typs 27256 mitgebrannt wird. Die Platine wird mit vergoldeten Anschlüssen samt Diskette, die die Software zum erzeugen der EPROM-Files enthält, sowie ausführlicher Bedienungsanleitung geliefert.

Die 128 KByte EPROM-Karte

Diese 'normale' EPROM-Karte kann bis zu vier EPROMs des Typs 27256 oder 27128 oder 2764 aufnehmen. Sie eignet sich somit für kleinere Programme.

Verlängerungskabel zum Easyprommer

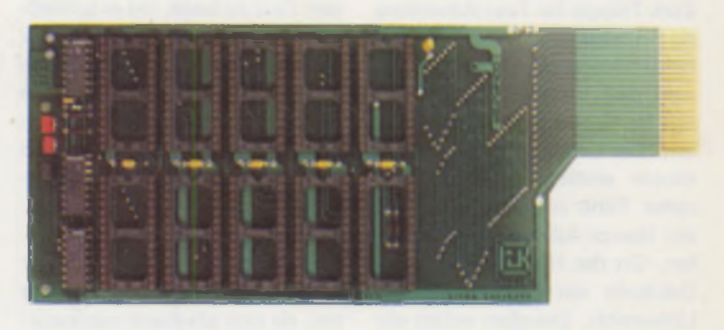
Mit diesem Kabel kann der Easyprommer auch neben dem Rechner betrieben werden. Dieses Kabel ist notwendig, wenn der Easyprommer im Gehäuse geliefert wird.

Software-Update zum Easyprommer!

Kunden, die eine ältere Software-Version zum Easyprommer besitzen, erhalten bei Einsendung der Original-Diskette und frankierten Rückumschlages sowie eines Kostenbeitrages von DM 10,– die neueste Version V.2.2.

Der Easyprommer V2.2

Ein universelles Programmiergerät für den ATARI ST, das nicht nur alle gängigen EPROMs der 27...-Serie inkl. 27011 (Megabit-Eprom) brennt, sondern auch die modernen EEPROMs lesen und programmieren kann. Die sehr komfortable Software, natürlich voll GEM unterstützt, erlaubt alle nur denkbaren Manipulationen. Mit dem integrierten HEX/ASCII-Monitor lassen sich Änderungen der Daten blitzschnell realisieren. Der auf der Platine befindliche Druckertreiber erlaubt das problemlose Drucken der Daten. Zwei Zusatzprogramme ermöglichen das Brennen **jeder** nicht kopiergeschützten Software direkt von Diskette einschließlich Accessories und Autostart-Programmen. Der Easyprommer wird an den Druckerport angeschlossen und samt Diskette und Handbuch geliefert.



Hiermit bestelle ich:

☐ Easyprommer (Fertiggerät) für DM 349,–

☐ Easyprommer (Bausatz) für DM 298,–

☐ 576 KByte EPROM-Karte (fertig bestückt o. EPROMs) für DM 148,–

☐ 128 KByte EPROM-Karte (fertig bestückt o. EPROMs) für DM 58,–

☐ Verlängerungskabel für Easyprommer für DM 39,90

Versandkosten: Inland DM 7,50 Ausland DM 10,–
Auslandsbestellungen nur gegen Vorauskasse

☐ Vorauskasse
☐ Nachnahme

Name: ____________ Vorname: ____________

Straße: ____________ Ort: ____________

Unterschrift: ____________

Bezugsquelle: MERLIN computer gmbh Computer GmbH
Schwalbacher Straße 64
D-6236 Eschborn
Tel. 0 61 96/ 48 21 58

Relax

The Lurking Horror und Stationfall

Neues für Infocom-Freunde

Nach Douglas Adams geradezu genialem 'Bureaucracy' sind 'The Lurking Horror' und 'Stationfall' die nächsten, wirklich sehr guten und anspruchsvollen Text-Adventures aus dem Hause Infocom. Dieser Name steht schon seit der legendären Zork-Trilogie für Text-Adventure an Spitzenklasse. Auch mit den beiden Neuerscheinungen zeigt sich, daß dieser Ruf gerechtfertigt ist.

Mit 'The Lurking Horror' hat Infocom endlich dem Wunsch vieler Fans nachgegeben und ein Horror-Adventure geschaffen. Ort der Handlung sind die Gebäude der fiktiven G.U.E. Universität. Draußen wütet ein entsetzlicher Schneesturm und der Spieler ist mehr oder weniger eingesperrt, was nicht von Nachteil ist, da man noch 20 Seiten für eine wichtige Klausur zu schreiben hat. Kaum hat man sich jedoch an seinen Terminal gesetzt und hat, nach Eingabe des Codewortes, das nur im Handbuch steht (Der beste Kopierschutz für die nicht geschützen Programme von Infocom sind immer noch die Handbücher.), angefangen den bisher getippten Text zu lesen, geht der Horror auch schon los. Auf dem Bildschirm finden sich Daten aus einem Alchemisten Labor in der Uni ein. Da geht es um Geisterbeschwörung und schwarze Magie. Geheimnisvolle Kräfte zwingen den Studenten den Text zu lesen, bis er schließlich in einen erschreckend realistischen Alptraum fällt. Nach dem er aufgewacht ist, sind alle Daten gelöscht und er muß sich erneut auf die Suche danach machen. Von einem Hacker bekommt er einen wertvollen Tip. Der Spieler muß jetzt in die Keller der Uni vorstoßen, in die er sich bisher noch nie gewagt hat, da sich allerhand merkwürdige Geschichten darüber erzählt werden. Aber nicht nur im Keller lernt man das Gruseln. Ein Mann der eine Bodenwachsmaschine in einem der Korridore fährt, stellt sich bei genauerer Untersuchung als Zombie heraus. Aber auch hier setzt sich das relativ gewaltlose Prinzip von Infocom durch, es hat wenig Sinn zu versuchen den Zombie mit roher Gewalt umzubringen.

Wenn man das Spiel länger spielt, breitet sich eine gruselige Atmosphäre aus, die bei dem Grusel-Adventure 'Moonmist' nicht so recht aufkommen wollte. Leute mit schwachen Nerven sollten 'The lurking Horror' also nicht zu später Stunde spielen. Der Author Dave Lebling hat ein Programm geschaffen, das einen Ehrenplatz in seiner bisherigen Erfolgsliste einnehmen kann. Er schuf Hits wie 'Spellbreaker', 'Starcross' und 'Suspect'. An der Zork-Trilogie war er als Co-Author beteiligt.



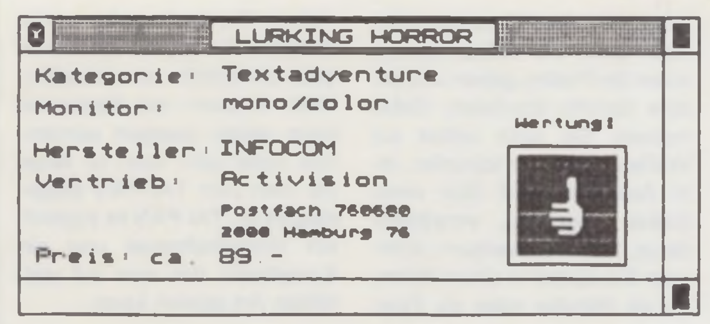
Die üblichen Nachteile bei Spielen von Infocom, Preis und Sprache sind auch hier wieder gegeben. Der Preis ist vielleicht durch das neue Design der Verpackung und die Zugaben in der Packung (außer Anleitung und Diskette) zu entschuldigen. Die Sprache, recht anspruchsvolles Englisch, gehört ebenfalls dazu und ist schon fast ein Markenzeichen. Aber mit durchschnittlichen Englischkenntnissen und einem dicken Wörterbuch läßt sich 'The Lurking Horror' wie auch alle anderen Infocom-Adventure recht flüssig spielen. Außerdem, und das sei an die Adresse der Schüler gerichtet, lernt man, wenn man mehrere Infocom-Adventures spielt, geradezu spielend Englisch.

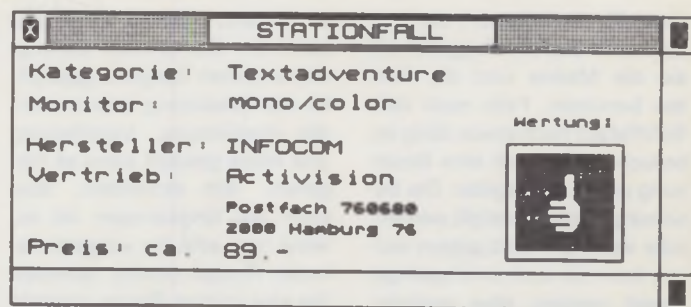
Möchte man etwas Anspruchsvolles auf den Bildschirm bekommen , dem sei 'The lurking Horror' wie auch jedes andere Infocom-Adventure empfohlen.



Zum Abschluß noch ein paar Tips für die Anfänger:

- Der Hacker rückt den entscheidenden Schlüssel nur raus, wenn er etwas vernünftiges zu Essen bekommt.
- Der Zombie hat Schwierigkeiten auf gewachstem Boden zu laufen.
- Cola ist koffein-haltig. Und das hat aufmunternde Wirkung.
- Das Monster, welches die Hand begiert, hat Angst vor dem Stein.
- Die Ratten 'beruhigt' man, indem man das Ventil am Rohr im richtigen Augenblick aufschlägt.
- Den Abschiedsbrief des Studenten zeigt man dem Professor.





Auch mit 'Stationfall' hat Infocom bzw. Steve E. Meretzky einen langersehnten Wunsch der großen Fangemeinde von 'Planetfall' erfüllt. Vor vier Jahren brachte Infocom das Science Fiction Adventure 'Planetfall' heraus. Nicht nur wegen des sehr guten Begleitcharacters, der je in einem Adventure von Infocom vorkam, nämlich dem genial kindischen, verspielten und urkomischen Roboter Floyd, forderten die Fans schon bald einen Nachfolger. Die lange Wartezeit hat jetzt ein Ende. Nachdem sie den Planeten Resida in 'Plantfall' vor dem Untergang gerettet haben, sind sie bis zum Leutnant erster Klasse befördert worden. Das Leben sollte jetzt aufregender werden, als es bisher als Raum(schiff)-pfleger war. Aber sie haben nur eine dieser Arbeiten gegen eine andere eingetauscht.

Jetzt müssen sie Schreibtischarbeit verrichten, bei der es das Aufregendste ist, wenn sie einen Ausflug zur Raumstation Gamma Delta Gamma 777-G 59/59 machen dürfen, um sich um Formulare zu kümmern.

Und das ist genau das, was sie heute tun müssen. Für alles, was sie hierzu brauchen, werden natürlich, wie könnte es anders sein, Formulare benötigt, die auch in der Packung enthalten sind.

Sind sie nach dem freudigen Wiedersehen mit Floyd endlich bei der Raumstation Gamma Delta Gamma 777-G 59/59 angekommen, wird es schon merkwürdig. Kein Empfangskommitee, nichts.

Die gesammte Station ist verlassen. Sie bekommen Hinweise auf eine Pyramide, die die ganze Menschheit bedroht. Ihre Aufgabe ist es nun, das Universum zu retten.

Im Laufe der Zeit zeigen sich die drastischen Auswirkungen der Pyramide. Zum Beispiel fängt Floyd an, sich sonderbar zu benehmen (unter anderem rülpst er und entschuldigt sich nicht dafür). Oder es funktionieren Maschinen nicht mehr wie erwartet und verschwinden sogar.

Am Ende des Spiels, so traurig es ist, muß die Fangemeinde für immer von Floyd abschiednehmen. Man bekommt einen Vertreter namens Oliver.

Für einen Teil des Spiels entfällt das Kartenzeichnen. Ein Plan der Raumstation liegt bei. Die angedockte Raumstadt darf der eifrige Abenteurer aber wie immer mit Papier und Bleistift erkunden.

Stationfall ist gagreicher als 'The Lurking Horror'. Schon allein die Wiedereinführung der Fußnoten, 'Hitchhiker'-Fans werden sich erinnern, und der Kommentar des Authors dazu ist sehenswert. Ebenso wird die Bürokratie auf die Schippe genommen.

Die meisten Probleme sind leicht zu lösen. Ein paar harte Nüsse sind aber auch noch da, die ja das Salz in der Suppe ausmachen.

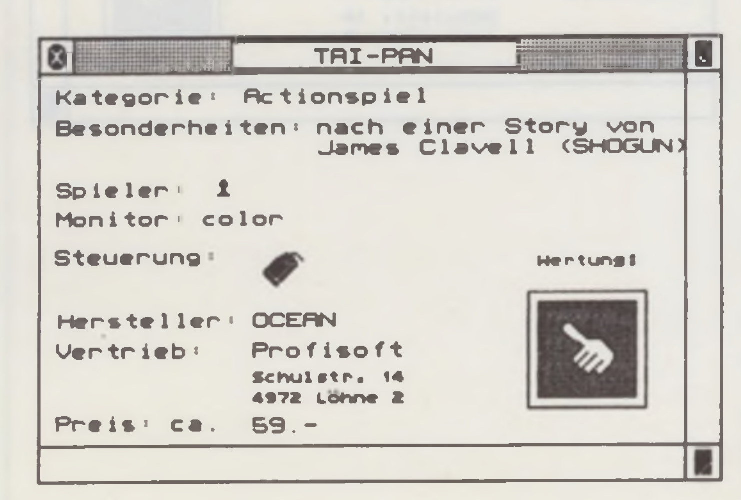
'Stationfall' ist ein unbedingtes Muß für alle begeisterten 'Planetfall' Spieler. 'Stationfall' ist sicher eines der Besten, wenn nicht das Beste Programm, daß Infocom je herausgebracht hat.

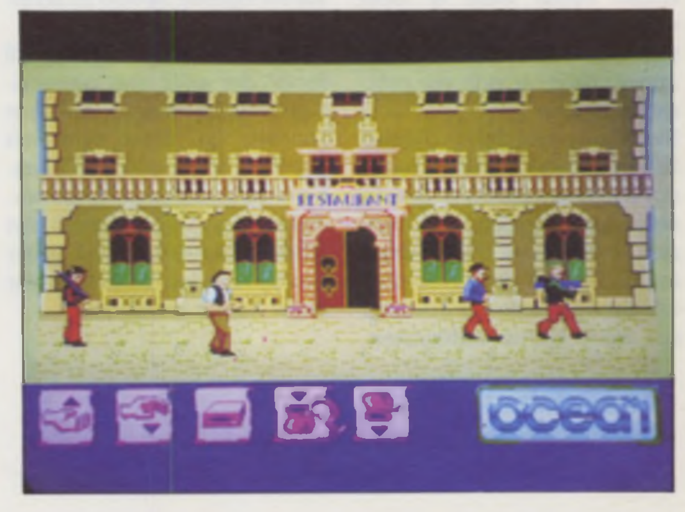
Andreas Paul

Hier noch ein paar Tips für Anfänger:

- Das zerknitterte Formular muß man pressen.
- Den Strauss lockt man mit Futter zu dem streikenden Automaten und erschrickt ihn.
- Der Aufbewahrungsort des Futters ist in einem Brief beschrieben.
- Die seltsamme Kreatur lockt man mit Hilfe der Spraydose in die Kapelle, wo man mit dessen Hilfe an den Stern herankommt.
- Die Thermosflasche isoliert auch gegen Wärme.

TAI-PAN
Gefährlicher Handel im Ocean





TAI-PAN ist ein Wirtschaftsspiel, welches sich Anfang des 19. Jahrhunderts in China abspielt. Sie beginnen nun das Spiel ohne einen Pfennig Geld. Ihre Aufgabe ist TAI-PAN zu werden. Zuerst müssen Sie jemanden finden der ihnen Geld leiht, um ein Schiff zu kaufen und auszurüsten. Es gibt drei Arten von Schiffen: die 'Lorcha', ein schnelles Schmugg-

lerschiff, den 'Klipper', ein Handelsschiff und die 'Fregatte' wie sie die Marine und die Piraten benutzen. Falls nach dem Schiffskauf noch etwas übrig ist, brauchen Sie noch eine Besatzung und Handelsgüter. Die Besatzung kann angestellt werden, oder wenn Sie Geld sparen wollen, kann sie auch zwangsangeheuert werden, aber vorsicht, dafür kann man ins Gefängnis kommen. Bei dreimal Gefängnis wird man übrigens geköpft. Ist die Besatzung beisammen, die Ausrüstung, Verpflegung und Ware gekauft kann es losgehen. Am sichersten, aber auch am langsamsten ist es, wenn man eine der vorgeschriebenen Routen nimmt. Nehmen Sie eine andere Route, besteht die Gefahr eines Piratenangriffs. Aber auch Sie selbst können unter die Piraten gehen und andere Schiffe überfallen. Dabei müssen Sie auch selbst zur Waffe greifen und kämpfen. Ihre Ausdauer wird über einen Balken angezeigt, verschwindet er, haben Sie verloren. Kommen Sie einmal in einen Hafen, ob als Händler oder als Pirat, müssen Sie das Lagerhaus oder die Bank finden, um die Waren oder die Schiffe zu verkaufen. Nach erneuern des Bestandes kann weiter gesegelt werden. Das Spiel geht dann so lange bis man zum TAI-PAN aufgestiegen ist. TAI-PAN ist zugleich ein Wirtschaftsspiel und ein Kampfspiel, das man auf vielfältige Art spielen kann.

KINGS QUEST III

Die Begegnung mit dem Zauberer

Vor langer Zeit lebte im Land von Llewdor ein Zauberer mit den Namen Manannan. Um sich nicht von unwichtigen Dingen aufhalten lassen zu müssen, besorgte sich der Zauberer einen kleinen Jungen, der für ihn die unwichtigen Dinge erledigte. Doch der Junge wuchs heran und abenteuerlustig wie er war, erkundete er die alten Bücher des Zauberers. Eines Tages beim Ausprobieren eines Zauberspruches erwischte ihn der Zauberer. Er war darüber so erbost, daß er ihn in ein Häufchen Staub verwandelte. Erschreckt über seine Unbeherrschtheit beschloß er, den selben Fehler wolle er nicht mehr machen. Bei seinem nächsten Sklaven, wieder ein kleiner Junge, nahm er sich vor achtsamer zu sein.

Sie müssen nun diesen Jungen führen und ihn aus den Fängen des Zauberers befreien. Das ist nicht ganz einfach, hat man zum Beispiel das Haus verlassen und ist auf dem Weg über das Gebirge, kann der Zauberer aus dem Nichts auftauchen. Dabei erscheint erst eine Wolke und dann der Zauberer selbst. Als Strafe wird man dann ins Haus zurückgezaubert. Schlimmer ist es, wenn man das zweite Mal auf der Flucht erwischt wird, als Strafe führt man dann merkwürdige Verrenkungen aus. Ist man dann in der glücklichen Lage dem Zauberer entkommen zu sein, beginnt der gefahrenvolle Gang durchs Gebirge, um dann an die See und an ein Schiff zu gelangen. Danach stehen noch gefahrvolle Begegnungen mit Medusa, die Menschen in Stein verwandeln kann und einem Schneemenschen an, um am Ende den Zauberer mit seinen eigenen Mitteln zu schlagen, nämlich der Magie.

Wer schon Kings Quest I und II kennt, der weiß, daß immer einige Überraschungen auftreten können. Die Figur kann mit Joystick, Maus oder der Tastatur in verschiedenen Geschwindigkeiten gesteuert werden. Ein schönes Adventure mit guter Grafik, nur das Nachladen der Bilder dauert etwas lang.







Nachdem Sie mit Sicherheit den Aufbau der Easytizer-Hardware erfolgreich absolviert haben, können Sie nun das 'harte' Werkzeug aus der Hand legen und das 'weiche' Werkzeug zur Hand nehmen. Dieser Artikel soll Ihnen zum Einen die genaue Ansteuerung des Easytizers nahebringen, und zum Anderen eine flexible Assemblerroutine vorstellen, die Sie z. B. in Ihre eigene Software einbringen können.

Ein kurzer Rückblick

Nachdem sich der letzte Beitrag ausführlich mit der Hardware des Easytizer's beschäftigt hat, wird der Easytizer in diesem zweiten und letzten Teil ausschließlich aus dem Blickpunkt der Software betrachtet. Zum Einen wird die Programierung auf Maschinenebene erläutert, und zum Anderen wird eine flexible Assemblerroutine vorgestellt, die Sie in Ihre eigene Software einbinden können, wie hier zum Beispiel in ein GFA-Basic Programm. So bleibt es letztendlich ihrer Phantasie überlassen, ob der Easytizer eine kleine und sinnvolle Ergänzung ihres Zeichenprogramms, oder eine 'Animationsmaschine' wird.

Ein wenig Wiederholung

Die Ansteuerung der Easytizer-Hardware wurde im vorigen Artikel bereits beschrieben, doch soll dies kurz wiederholt werden. Durch die Easytizer-Hardware wird der ROM-Bereich von $FB0000 -$FBFFFF belegt. In diesem 64K Block wiederholen sich periodisch die Register des Easytizers, über die der Datenaustausch mit dem Rechner stattfindet. Da das Schreiben in diesem Bereich nicht möglich ist, (sonst 2 Bömbchen – Busfehler) Betriebszustände der Hardware jedoch durch den Rechner manipulierbar sein sollen, wird einfach der Zugriff auf ganz bestimmte Register als Ausgabe interpretiert. Außerdem sollten nur Wortzugriffe oder Bytezugriffe auf geraden Adressen stattfinden, da sich die Hardware sonst nicht ansprechen läßt.

$FB0000 SR0, nur Wort:
In diesem Register sind 16 Bildpunkte des niederwertigen Teils der Bildinformation (Bitplane 0) verfügbar. Der Inhalt dieses Registers stammt aus einem Schieberregister, in welches alle 4 Microsekunden die 16 Bildpunkte der momentan über den Spaltenzähler eingestellten Spalte übertragen werden. Ein Zugriff auf diese Register bewirkt gleichzeitig ein Zurücksetzen der LWF-Flagge im Statusregister.

$FB0002 SR1, nur Wort:
Wie oben, jedoch der höherwertige Teil der Bildinformation (Bitplane 1). Beide Register sind so organisiert, daß sie direkt mit dem Bildspeicher des Atari korrespondieren, und deshalb nur zum richtigen Zeitpunkt an die richtige Stelle des Bildspeichers kopiert werden müssen, damit auf dem Monitor das digitalisierte Abbild des Videosignals entsteht.

$FB0004 INZ, Wort/Byte:
Ein Zugriff auf dieses Register inkrementiert den sog. 'Spaltenzähler', mit dem kontrolliert wird, welche von vier Spalten zu 16 Bildpunkten in die oben genannten Register übertragen werden. Die gelesenen Daten enthalten hier keine Information, nur der Zugriff zählt. Es sind vier Zugriffe nötig, um die Spalte 16 Bildpunkte nach rechts zu rücken.

$FB0006 LZ, Wort/Byte:
Ein Zugriff auf dieses Register löscht den Spaltenzähler. Damit ist die am weitesten links liegende Spalte eingestellt.

$FB0008 STATUS, Wort/Byte:
Status-Register, das über Synchronsignale im Easytizer Auskunft gibt. Von 16 Bits werden lediglich vier benutzt:

Bit 15: LWF
Diese Flagge wird gesetzt, wenn ein neues Langwort aus den Schieberegistern IC1–IC4 in die intern nachgeschalteten Register übertragen wurde und wird durch einen Zugriff auf SR0 und am Anfang einer Zeile gelöscht. Zweck dieser Flagge ist es, eine Synchronisation zwischen Rechner und Videosignal zu ermöglichen. Aus Zeitgründen ist es jedoch nur möglich, am Anfang einer Zeile auf den Spaltentakt zu Synchronisieren. Probleme ergeben sich jedoch nicht, da die Taktfrequenz des ATARI-ST (genau 8 Mhz) ein Vielfaches des Spaltentaktes (genau 250 Khz) ist und bei richtigem 'Timing' der Software keine Zeitverschiebungen auftreten.

Bit 14: CSYNC
Das pure, positive Synchronsignal, über ein Monoflop um etwa 7,8 Microsekunden verlängert, das ebenfalls zur Synchronisation des Rechners mit dem Videosignal dient.

Bit 13: VSYNC
Dieses Bit wird für etwa 160 Microsekunden Dauer während eines Rasterwechsels aktiv.

Bit 12: TASTE
Dieses Bit wird bei gedrückter 'Schnappschuss-Taste' aktiv.

Das Zeitproblem:

Obwohl der ATARI-ST nicht einer der müdesten Rechner ist, kann er ohne DMA (Speicherdirektzugriff) oder ähnliche Hardwareintensive Einrichtungen nicht ohne Weiteres die, bei der Videobildverarbeitung anfallenden, Datenmengen in Echtzeit aufnehmen. Deshalb mußte der Datenberg in vier kleinere, verdaulichere Portionen aufgeteilt werden. Also wurde eine Videozeile in vier Spalten zu je 16 Bildpunkten aufgeteilt, die sich im Abstand von 64 Bildpunkten wiederholen. Natürlich erkauft man sich keinen Vorteil ohne Nachteile, denn um das vollständige Videobild in den Rechner zu übertragen, benötigt man jetzt die vier- bzw. achtfache Zeit (80/160 ms). Somit muß die Software nur noch im Abstand von 4 Microsekunden ein Langwort aus dem Easytizer an die richtige Stelle des Bildspeichers transportieren.

Der Punkttakt beträgt 16 Mhz (62.5 ns/Punkt), also würde ohne eine Spalteneinteilung die Spaltenfrequenz, also die Frequenz, mit der jeweils 16 neue Bildpunkte in den Schieberregistern bereitstehen, genau 1 Mhz (16 Mhz/16 Bildpunkte) betragen. Abgesehen davon, daß sich mit dieser Frequenz kein softwaremäßiger Datentransfer mehr abwickeln läßt, ergibt sich noch ein weiteres Problem: Das Zeitfenster, das unweigerlich beim Synchronisieren mit dem Spaltentakt entsteht, würde größer sein, als der Spaltenabstand. Das sogenannte Zeitfenster kommt dadurch zustande, daß der Zustand dieses Bit's periodisch in einem bestimmten Zeitabstand (hier 2.5 Microsekunden) abgetastet wird und daher, daß beide Vorgänge asynchron zueinander ablaufen, es zu einer maximalen 'Streuung' kommen kann, die der Abtastzeit entspricht. Dieses Zeitfenster (oder auch Phasenstreuung) beträgt mit allen

Programmiertricks etwa 2.5µs, also liegt man mit 4 Microsekunden Spaltenabstand (Spaltenfrequenz 250 Khz) auf der sicheren Seite.

Praktische Möglichkeiten des Easytizer's

Der Easytizer ist in seinen Anwendungen nicht durch die Grafikmöglichkeiten des ATARI-ST beschränkt, weshalb es möglich ist, ein Bild mit 640★200 Punkten und 4 Graustufen direkt in den Bildspeicher zu digitalisieren, aber auch mit einer Auflösung von 768★600 Punkten und vier Graustufen zu digitalisieren, wenn es auch nicht direkt sichtbar wird. Anwendungen, die nur eine Graustufe (monochrome Bilder) erfordern, kann der Easytizer genauso erfüllen, wie Anwendungen, in denen vier Graustufen nötig sind. Alle Möglichkeiten werden von dem Assemblermodul Easysoft unterstützt, auf das später genauer eingegangen werden soll.

Falls Sie, aus welchen Gründen auch immer, nicht dieses Assemblermodul benutzen wollen, sondern Ihre eigene Software zur Ansteuerung des Easytizer's schreiben wollen, sei Ihnen so viel gesagt: Es soll Ihnen keine Information über den Easytizer vorenthalten werden, jedoch können wir hier auch nicht auf jede Einzelheit ausführlich eingehen und deshalb soll sich der Artikel auf die Benutzung der Easysoft-Routinen konzentrieren. Wir bieten Ihnen das hier abgedruckte Assemblermodul an, mit dessen Möglichkeiten wohl jede Anwendung abgedeckt werden kann.

Der Grafikaufbau des ST

soll hier kurz angeschnitten werden, da er unmittelbar in Verbindung mit dem Easytizer steht. Der Grafikspeicher des ST besteht aus einem Speicherblock, der eine bis vier Bildebenen umfaßt. Die Anzahl der Bildebenen (Bitplanes) bestimmt die Anzahl der möglichen Farben pro Bildpunkt und hängt von der eingestellten Auflösung ab. So ist es in der monochromen Auflösung nur eine Bildebene, in der mittleren Auflösung sind es zwei und in der niedrigeren sind es vier Bildebenen. In der monochromen Auflösung bilden 40 Speicherworte eine komplette Bildschirmzeile, wobei das Wort mit der niedrigsten Adresse, die am weitesten links liegenden Bildpunkte beinhaltet. Ein Wort (Wort = 16 Bits) bestimmt die Farbe (hier: hell/dunkel) von jeweils 16 Bildpunkten, und zwar ist Bit 15 eines Speicherwortes dem am weitesten links liegenden Bildpunkt zugeordnet. Nach jeweils 40 Speicherworten fängt eine neue Zeile an. In anderen Auflösungen gilt dies auch, mit der Ausnahme, daß jeweils zwei bzw. vier aufeinanderfolgende Worte zu 16 Bildpunkten zusammengefaßt werden, und dadurch zwei bzw. vier Bits pro Bildpunkt und damit entsprechend mehr Farben pro Bildpunkt möglich werden. Genau in diesem Format sind auch die Speicherworte vom Easytizer verfügbar, und müssen deshalb nur noch zur richtigen Zeit an die richtige Stelle in den Bildspeicher kopiert werden.

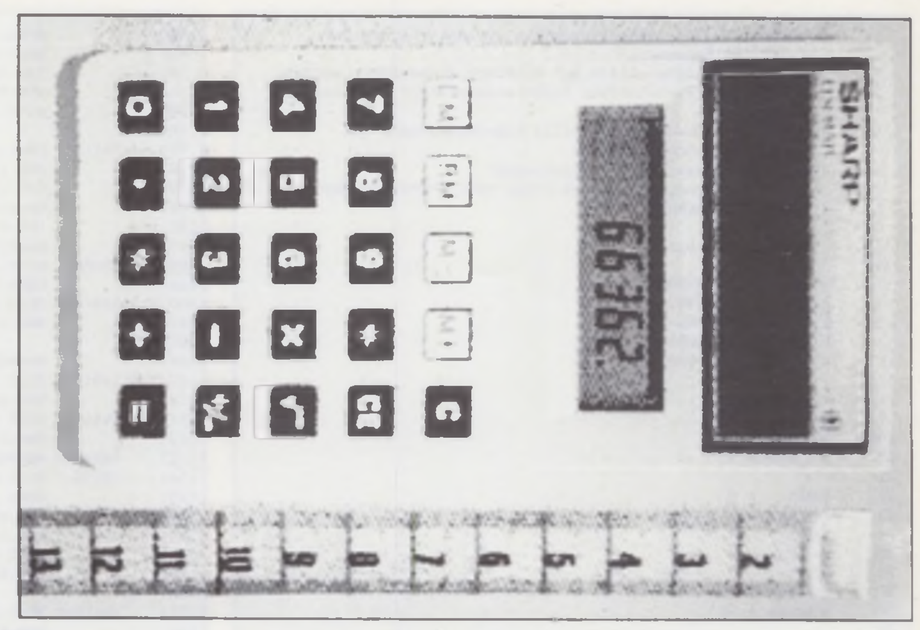


Bild 1: Digitalisierung mit vier Graustufen in mittlerer Auflösung

Die Ansteuerung des Easytizer's

ist aus Geschwindigkeitsgründen nur auf Maschinenebene möglich.

Ein Assemblerprogramm, das den Easytizer ansteuert (auch Easysoft) sollte nach folgenden Schritten ablaufen:

1. Löschen des Spaltenzählers.
2. Den Bildwechsel des Videosignals abwarten. (Bit VSYNC im Statusregister)
3. Bis zur gewünschten ersten Zeile des Videosignals warten.
4. Synchronisation mit den Schieberregistern über das Bit LWF im Statusregister.
5. Die Spalte der Bildinformation aus den Schieberegistern an die entsprechende Position des Bildspeichers übertragen.
6. 4 Microsekunden warten, bis die nächste Spalte in die Schieberegister übertragen ist.
7. Ab Schritt 5 solange wiederholen, bis alle Spalten einer Zeile übertragen sind.
8. Ab Schritt 4 solange wiederholen, bis alle Zeilen des Halbbildes übertragen sind.
9. Spaltenzähler durch viermaligen Zugriff auf Register INZ auf die nächste Spalte einstellen.
10. Ab Schritt 2 solange wiederholen, bis alle Spalten übertragen sind.
11. Fertig.

Verzweifeln sie nicht, denn all diese Bitzwirbeleien nimmt Ihnen das Programmodul EASYSOFT ab, ohne das Sie sich je mit Assemblerprogrammierung auseinandersetzen müssen. Hier wird die Benutzung der EASYSOFT-Routinen aus GFA-Basic heraus beschrieben, doch läßt sich das genausogut für 'C' oder Pascal realisieren.

EASYSOFT – find' ich gut

In Listing 1 sehen Sie die EASYSOFT-Routinen abgedruckt. Es werden 3 Funktionen bereitgestellt; die Parameterübergabe erfolgt nach 'C'-Konventionen auf dem Stack. Die Routinen sind vollständig PC-Relativ (Program Counter) geschrieben und damit an jeder Stelle des Speichers lauffähig.

```
  1:    ; EASYSOFT - Die Assemblerroutine zur Ansteuerung der
          EASYTIZER Hardware.
  2:    ; Dieser Quellcode sollte mit Profimat assembliert werden,
  3:    ; wobei weder PC-Relativer noch relozierbarer Code erzeugt
          werden darf.
  4:    ; Das Programmmodul ist voellig Lageunabhaengig und
          benoetigt keinen weiteren
  5:    ; Speicher ausser seiner Eigenlaenge.
  6:    ; Parameteruebergabe auf dem Stack nach C-Konventionen.
  7:    ; Raymund Hofmann 21/9/87
  8:
  9:      ilabel tos\tos.l
 10:
 11:    h_sr0    = $fb8000
 12:    h_sr1    = $fb8002
 13:    h_inz    = $fb8004
 14:    h_lz     = $fb8006
 15:    h_status = $fb8008
 16:
 17:    b_lwf    = 7
 18:    b_csync  = 6
 19:    b_vsync  = 5
 20:    b_taste  = 4
 21:
 22:    funkt    = 0
 23:    xweite   = 2
 24:    yweite   = 4
 25:    interl   = 6
 26:    bplanes  = 8
 27:    anfang   = 10
 28:    adresse  = 12
 29:
 30:    trabant  = 5
 31:    nil      = $23330444
 32:
 33:    einsprung:
 34:            lea einsprung(pc),a6    ; Basisadresse
 35:            lea 4(sp),a5            ; Parameterbasis
 36:            move.l a5,parbase(a6)   ; Sichern
 37:            tst.w funkt(a5)         ; Welche Funktion ?
 38:            beq.s v_foto
 39:            cmp.w #1,funkt(a5)      ; Taste ?
 40:            beq.s taste
 41:
 42:            lea b_status,a0         ; Testen auf Videosignal
 43:            move.b (a0),d0
 44:            and.b #%01000000,d0     ; Csync-Bit herausfiltern
 45:            move.w #100,d1          ; Zeitlimit
 46:    \testv  move.b (a0),d2
 47:            and.b #%01000000,d2
 48:            eor.b d0,d2             ; Hat das Bit seinen Zustand
                                          behalten ?
 49:            dbne d1,\testv          ; JA
 50:            moveq #0,d0
 51:            cmp.w #-1,d1            ; Timeout
 52:            beq.s \kvid             ; Ja - Kein Videosignal
 53:            moveq #1,d0
 54:    \kvid   rts
 55:
 56:    taste   moveq #0,d0
 57:            btst #b_taste,b_status  ; Taste gedrueckt ?
 58:            beq.s ntast
 59:            moveq #1,d0
 60:    ntast   rts
 61:
 62:    v_foto  IKBDWS ikbdstp(pc),#0   ; Ikbd zum Schweigen bringen
 63:            move.w #1000,d0
 64:    \zeit   nop
 65:            dbra d0,\zeit           ; 2 ms warten
 66:            SUPEXEC easy(pc)        ; Im Supervisormodus ausf.
 67:            IKBDWS ikbdsta(pc),#0   ; Redeverbot aufheben
 68:            moveq #0,d0             ; Immer fehlerfrei
 69:            rts                     ; Rueckkehr
 70:
 71:    easy    move.l parbase(pc),a5   ; Parameterliste
 72:            move sr,save_sr(a6)     ; Statusregister retten
 73:            ori #$0700,sr           ; Interrupts ausmaskieren
 74:            st flagge(a6)           ; Flagge setzen
 75:            lea h_sr0,a0            ; Niederwertige Bitebene
 76:            lea h_sr1,a1            ; Hoeherwertige Bitebene
 77:            lea h_status,a2         ; Statusregister
 78:            moveq #b_csync,d0       ; Synchronisations-Bit
 79:            clr.w d2                ; Spalte im Bildspeicher
 80:            moveq #12,d1            ; maximal 12*64 Bildpunkte
 81:            sub.w xweite(a5),d1
 82:            bpl.s \ugok             ; gross genug
 83:            clr.w d1                ; Loeschen
 84:    \ugok   cmp.w #11,d1            ; Zu gross ?
 85:            bls.s \ogok             ; Nein
 86:            moveq #11,d1            ; Ja
 87:    \ogok   move.w d1,d3
 88:            mulu #sprung2-sprung1,d1; mal Laenge des Programmteils
 89:            lea sprung1(pc),a3      ; Adresse des 1.Programmteils
 90:            add.w d1,a3             ; plus Verschiebung
 91:            move.l a3,spadr+2(a6)   ; in den JMP-Befehl einsetzen
 92:            mulu #sprung4-sprung3,d3; mal Laenge
 93:            lea sprung3(pc),a3
 94:            add.w d3,a3
 95:            move.l a3,spadr2+2(a6)  ; in 2. JMP-Befehl einsetzen
 96:
 97:    lbild   btst #b_vsync,(a2)      ; Bildwechsel aktiv ?
 98:            bne.s lbild             ; ja
 99:            tst.w h_lz              ; Spaltenzaehler loeschen
100:            move.w xweite(a5),d1    ; Horizontale Aufloesung
101:            lsl.w #3,d1             ; *8=Bytes pro Spalte
102:            moveq #0,d7             ; Halbbildzaehler
103:    bildper move.l adresse(a5),a3   ; Adresse des Bildspeichers
104:            adda.w d2,a3            ; Spaltenposition aufaddieren
105:    \wartbw btst #b_vsync,(a2)      ; Warten auf den Bildwechsel
106:            beq.s \wartbw
107:
108:            moveq #trabant,d4       ; Trabantenzaehler
109:    \whics  btst d0,(a2)            ; Csync high ?
110:            beq.s \whics            ; Nein
111:    \wlocs  btst d0,(a2)            ; Csync low ?
112:            bne.s \wlocs            ; Nein
113:    \messen moveq #100,d5
114:    \mslp   btst d0,(a2)
115:            dbne d5,\mslp           ; D6 dekrementieren, sol. low
116:            cmp.w #100-16,d5        ; Lowzeit > 40 microsec ?
117:            bcs.s \weiter
118:            subq.w #1,d4            ; Ein Trabant weniger ...
119:            bra.s \wlocs
120:
121:    \weiter btst d0,(a2)            ; Csync low ?
122:            bne.s \weiter           ; Nein
123:            move.w anfang(a5),d5    ; 1. Zeile
124:    \zeil   btst d0,(a2)            ; komplette Zeile abwarten
125:            beq.s \zeil
126:    \wzhi   btst d0,(a2)
127:            bne.s \wzhi
128:            dbra d5,\zeil           ; Bis zur ersten Zeile
129:    \syncbj btst d0,(a2)
130:            bne.s \syncbj
131:
132:            tst.w bplanes(a5)       ; 2 Bildebenen ?
133:            bne zweibe
134:
135:    ; Dieser Teil uebernimmt das Einlesen von einer Bildebene
136:
137:    einbe   move.w yweite(a5),d5    ; Zeilenzahl
138:            tst.w interl(a5)        ; Interlace aus ?
139:            bne.s \iaus
140:            lsr.w #1,d5             ; /2 , da Interlace
141:    \iaus   subq #1,d5
142:            tst.w d4                ; Welches Halbbild angesagt ?
143:            beq.s wazl              ; 1. Halbbild vorgezogen
144:            add.w d1,a3             ; Ungerade Zeilen
145:            tst.w interl(a5)        ; Interlace aus ?
146:            bne lbild               ; Ja, dann 2. Halbbild nicht
                                        ; auswerten
147:    wazl    btst d0,(a2)            ; Auf Anfang der Zeile warten
148:            beq.s wazl
149:    \synreg tst.b (a2)
150:            bpl.s \synreg
151:            tst.w (a0)
152:            tst.l (a0)
153:            tst.l (a0)
154:            tst.l (a0)
155:    spadr   jmp nil                 ; Adresse kommt spaeter
156:
157:    sprung1 move.w (a1),(a3)        ; 1. Spalte
158:            add.w #8,a3
159:            tst.w (a0)
160:    sprung2 move.w (a1),(a3)        ; 2. Spalte
161:            add.w #8,a3
162:            tst.w (a0)
163:            move.w (a1),(a3)        ; 3. Spalte
164:            add.w #8,a3
165:            tst.w (a0)
166:            move.w (a1),(a3)        ; 4. Spalte
167:            add.w #8,a3
168:            tst.w (a0)
169:            move.w (a1),(a3)        ; 5. Spalte
170:            add.w #8,a3
171:            tst.w (a0)
172:            move.w (a1),(a3)        ; 6. Spalte
173:            add.w #8,a3
174:            tst.w (a0)
175:            move.w (a1),(a3)        ; 7. Spalte
176:            add.w #8,a3
177:            tst.w (a0)
178:            move.w (a1),(a3)        ; 8. Spalte
179:            add.w #8,a3
180:            tst.w (a0)
181:            move.w (a1),(a3)        ; 9. Spalte
182:            add.w #8,a3
183:            tst.w (a0)
```

Listing 1: Der Easysoft-Quelltext sollte mit Profimat assembliert werden.

```
184:            move.w (a1),(a3)       ; 10. Spalte
185:            add.w #8,a3
186:            tst.w (a0)
187:            move.w (a1),(a3)       ; 11. Spalte
188:            add.w #8,a3
189:            tst.w (a0)
190:            move.w (a1),(a3)       ; 12. Spalte
191:            add.w #8,a3
192:            tst.w interl(a5)       ; Interlace ?
193:            bne.s \ioff            ; Nein
194:            add.w d1,a3            ; Eine Zeile weiterspringen
195:    \ioff   dbra d5,waz1           ; Und nächste Zeile einlesen
196:            addq.w #1,d7           ; Halbbildzaehler
197:            btst #0,d7             ; Vollbildperiode ?
198:            bne.s \nvoll
199:            tst.w h_inz            ; inkr. Spaltenzaehler
200:            tst.w h_inz
201:            tst.w h_inz
202:            tst.w h_inz
203:            addq.w #2,d2           ; 2 Bytes pro Spalte
204:    \nvoll  cmp.w #8,d7            ; Bild Vollstaendig ?
205:            bne bildper            ; Noch nicht
206:    ende    move.w save_sr(pc),sr
207:            rts
208:
209:    ; Dieser Teil uebernimmt das Einlesen von zwei Bildebenen
210:
211:    zweibe  move.w yweite(a5).d5   ; Zeilenzahl
212:            tst.w interl(a5)       ; Ohne Interlace ?
213:            bne.s \intl            ; Nein
214:            lsr.w #1,d5            ; /2 , da Interlace
215:    \intl   subq #1,d5
216:            tst.w d4               ; Welches Halbbild ist
                                       ; angesagt ?
217:            beq.s waz2
218:            cmp.w #1,interl(a5)    ; Interlace-Modus 1 ?
219:            beq bildper
220:            cmp.w #2,interl(a5)    ; Interlace-Modus 2 ?
221:            beq \spezial           ; Ja, dann beide Halbbilder
                                       ; nutzen
222:            add.w d1,a3            ; Ungerade Zeilen
223:            add.w d1,a3
224:            bra.s waz2
225:    \spezial tst.b flagge(a6)      ; Spezial Modus
226:            beq.s waz2             ; Schon synchronisiert
227:            clr.b flagge(a6)       ; Jetzt wird Synchronisiert
228:            bra bildper            ; Nur mit 1. Halbbild anfangen
229:
230:    waz2    btst d0,(a2)           ; Auf Anfang der Zeile warten
231:            beq.s waz2
232:    \synreg tst.b (a2)
233:            bpl.s \synreg
234:            nop
235:            tst.l (a0)
236:            tst.l (a0)
237:            tst.l (a0)
238:    spadr2  jmp nil                ; Adresse kommt spaeter
239:
```

```
240:    sprung3 move.l (a0),(a3)       ; 1. Spalte
241:            add.w #16,a3           ; Offset in naechste Spalte
242:    sprung4 move.l (a0),(a3)       ; 2. Spalte
243:            add.w #16,a3
244:            move.l (a0),(a3)       ; 3. Spalte
245:            add.w #16,a3
246:            move.l (a0),(a3)       ; 4. Spalte
247:            add.w #16,a3
248:            move.l (a0),(a3)       ; 5. Spalte
249:            add.w #16,a3
250:            move.l (a0),(a3)       ; 6. Spalte
251:            add.w #16,a3
252:            move.l (a0),(a3)       ; 7. Spalte
253:            add.w #16,a3
254:            move.l (a0),(a3)       ; 8. Spalte
255:            add.w #16,a3
256:            move.l (a0),(a3)       ; 9. Spalte
257:            add.w #16,a3
258:            move.l (a0),(a3)       ; 10. Spalte
259:            add.w #16,a3
260:            move.l (a0),(a3)       ; 11. Spalte
261:            add.w #16,a3
262:            move.l (a0),(a3)       ; 12. Spalte
263:            add.w #16,a3
264:            tst.w interl(a5)       ; Interlace aus ?
265:            bne.s \iaus            ; Ja, dann Zeile nicht
                                       ; ueberspringen
266:            add.w d1,a3            ; Eine Zeile weiterspringen
267:            add.w d1,a3
268:    \iaus   dbra d5,waz2           ; Und naechste Zeile einlesen
269:            addq.w #1,d7           ; Halbbildzaehler
270:            tst.w interl(a5)       ; Interlace aus ?
271:            beq.s \inaus           ; Nein
272:            addq.w #1,d7           ; Ja, nur 4 Halbbilder
273:    \inaus  btst #0,d7             ; Vollbildperiode ?
274:            bne.s \nvoll
275:            tst.w h_inz            ; inkr. Spaltenzaehler
276:            tst.w h_inz
277:            tst.w h_inz
278:            tst.w h_inz
279:            addq.w #4,d2           ; 4 Bytes pro Spalte
280:    \nvoll  cmp.w #8,d7            ; Bild Vollstaendig ?
281:            bcs bildper            ; Noch nicht
282:            bra ende
283:
284:
285:    ; Datenbereich
286:
287:    ikbdstp dc.b $13
288:    ikbdsta dc.b $11
289:    flagge  dc.w 0
290:
291:    parbase dc.l 0
292:
293:    save_sr dc.w 0
294:            end
295:
296:
```

Listing 1: Der Easysoft-Quelltext

```
 1:    ' Minitizer/Easytizer
 2:    ' Eine Demonstration zur Anwendung der Easysoft-Routinen.
 3:    ' Die Easysoft-Routinen werden aus den DATA Zeilen in
       Easy$ geladen und von da wie eine C-Funktion aufgerufen
 4:    ' ST-Computer 11/87
 5:    ' Raymund Hofmann 9/87
 6:    '
 7:    ' Standard Programmkopf
 8:    For I%=1 To 34
 9:      Read A%
10:      Header$=Header$+Chr$(A%)
11:    Next I%
12:    Void Xbios(&H6,L:Varptr(Header$)+2)
13:    Repeat
14:      Read A%
15:      B%=B%+A%
16:      Easy$=Easy$+Chr$(A%)
17:    Until A%=-1
18:    Read A%
19:    If A%<>(B%+1)
20:      Alert 1,"Fehler in den Data-|Zeilen.",1,"O.K.",Dummi
21:      End
22:    Endif
23:    Easytizer%=Varptr(Easy$)
24:    ' Hauptschleife
25:    '
26:    '
27:    Do
28:      @Dmenu
29:      Repeat
30:        In$=Inkey$
31:      Until Not In$=""
32:      @Udmenu
33:      Menu%=Instr(1,"DTLSE",Upper$(In$))
34:      On Menu% Gosub Digi,Trigger,Laden,Sichern,Ende
35:    Loop
36:    '
37:    '
38:    '
39:    Procedure Digi
40:      If C:Easytizer%(2)=1
41:        Void C:Easytizer%(0,Xmax/64,Ymax,Res,Res,50,L:Xbios(&H3))
42:      Else
43:        Alert 1,"Kein Videosignal.",1,"O.K.",Dummi
44:      Endif
45:    Return
46:    '
47:    '
48:    '
49:    Procedure Trigger
50:      If C:Easytizer%(2)=1
51:        Do
52:          Void C:Easytizer%(0,Xmax/64,Ymax,Res,Res,50,L:Xbios(&H
            3))
53:          Exit If Inkey$=Chr$(13) Or C:Easytizer%(1)=1
54:        Loop
55:      Else
56:        Alert 1,"Kein Videosignal.",1,"O.K.",Dummi
57:      Endif
58:    Return
```

Listing 2: Ein Beispielprogramm in GFA-Basic.

```
59:     '
60:     '
61:     '
62:     Procedure Laden
63:       Fileselect "\*.*","LADEN.pi3",Dat$
64:       If Not Dat$=""
65:         Get 0,0,Xmax-1,Ymax-1,Bild$
66:         Hd2$=Left$(Bild$,6)
67:         Bild$=Space$(32034)
68:         Bload Dat$,Varptr(Bild$)
69:         Header$=Left$(Bild$,34)
70:         Bild$=Hd2$+Right$(Bild$,32000)
71:         Put 0,0,Bild$
72:         Void Xbios(&H6,L:Varptr(Header$)+2)
73:       Endif
74:     Return
75:     '
76:     '
77:     '
78:     Procedure Sichern
79:       Fileselect "\*.*","SICHERN.pic",Dat$
80:       Get 0,0,Xmax-1,Ymax-1,Bild$
81:       If Not Dat$=""
82:         Header$=Chr$(0)+Chr$(Res)+Right$(Header$,32)
83:         Open "O",#1,Dat$
84:         Print #1,Header$;
85:         Print #1,Right$(Bild$,32000);
86:         Close #1
87:       Endif
88:       Put 0,0,Bild$
89:     Return
90:     End
91:     '
92:     '
93:     '
94:     Procedure Ende
95:       End
96:     Return
97:     '
98:     '
99:     '
100:    Procedure Dmenu
101:      Deffill 1,0,0
102:      @Haufl
103:      X1=0.27*Xmax
104:      Y1=0.74*Ymax
105:      X2=0.73*Xmax
106:      Y2=0.9*Ymax
107:      Get X1,Y1,X2,Y2,Save$
108:      Prbox X1,Y1,X2,Y2
109:      Texty=Y1+Ymax/25
110:      Deftext 1,16
111:      Text X1+50,Texty,"Minitizer/Easytizer"
112:      Deftext 1,0
113:      Text X1+16,Texty+Ymax/20,"D - Digitalisieren  L - Laden"
114:      Text X1+16,Texty+Ymax/10,"T - Trigger          S - Sichern"
115:    Return
116:    '
117:    '
118:    '
119:    Procedure Udmenu
120:      Put X1,Y1,Save$
121:    Return
122:    '
123:    '
124:    '
125:    Procedure Haufl
126:      Res=2-Xbios(&H4)
127:      If Res=0
128:        Xmax=640
129:        Ymax=400
130:      Else
131:        Xmax=640
132:        Ymax=200
133:      Endif
134:    Return
135:    ' Die Farbpalette
136:    '
137:    '
138:    Data &H00,&H00
139:    Data &H00,&H00,&H03,&H33,&H05,&H55,&H07,&H77
140:    Data 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
141:    ' Easysoft Routinen
142:    ' Entspricht dem Assembler Programm (Listing 1)
143:    '
144:    Data 77,250,255,254,75,239,0,4,45,77
145:    Data 2,232,74,109,0,0,103,64,12,109
146:    Data 0,1,0,0,103,40,65,249,0,251
147:    Data 128,8,16,16,192,60,0,64,50,60
148:    Data 0,100,20,16,196,60,0,64,177,2
149:    Data 86,201,255,246,112,0,178,124,255,255
150:    Data 103,2,112,1,78,117,112,0,8,57
151:    Data 0,4,0,251,128,8,103,2,112,1
152:    Data 78,117,72,122,2,144,63,60,0,0
153:    Data 63,60,0,25,78,78,80,143,48,60
154:    Data 3,232,78,113,81,200,255,252,72,122
155:    Data 0,30,63,60,0,38,78,78,92,143
156:    Data 72,122,2,107,63,60,0,0,63,60
157:    Data 0,25,78,78,80,143,112,0,78,117
158:    Data 42,122,2,90,64,238,2,236,0,124
159:    Data 7,0,80,238,2,230,65,249,0,251
160:    Data 128,0,67,249,0,251,128,2,69,249
161:    Data 0,251,128,8,112,6,66,66,114,12
162:    Data 146,109,0,2,106,2,66,65,178,124
163:    Data 0,11,99,2,114,11,54,1,194,252
164:    Data 0,8,71,250,0,162,214,193,45,75
165:    Data 1,106,198,252,0,6,71,250,1,126
166:    Data 214,195,45,75,2,84,8,18,0,5
167:    Data 102,250,74,121,0,251,128,6,50,45
168:    Data 0,2,231,73,126,0,38,109,0,12
169:    Data 214,194,8,18,0,5,103,250,120,5
170:    Data 1,18,103,252,1,18,102,252,122,100
171:    Data 1,18,86,205,255,252,186,124,0,84
172:    Data 101,4,83,68,96,234,1,18,102,252
173:    Data 58,45,0,10,1,18,103,252,1,18
174:    Data 102,252,81,205,255,246,1,18,102,252
175:    Data 74,109,0,8,102,0,0,204,58,45
176:    Data 0,4,74,109,0,6,102,2,226,77
177:    Data 83,69,74,68,103,8,214,193,74,109
178:    Data 0,6,102,138,1,18,103,252,74,18
179:    Data 106,252,74,80,74,144,74,144,74,144
180:    Data 78,249,35,51,4,68,54,145,214,252
181:    Data 0,8,74,80,54,145,214,252,0,8
182:    Data 74,80,54,145,214,252,0,8,74,80
183:    Data 54,145,214,252,0,8,74,80,54,145
184:    Data 214,252,0,8,74,80,54,145,214,252
185:    Data 0,8,74,80,54,145,214,252,0,8
186:    Data 74,80,54,145,214,252,0,8,74,80
187:    Data 54,145,214,252,0,8,74,80,54,145
188:    Data 214,252,0,8,74,80,54,145,214,252
189:    Data 0,8,74,80,54,145,214,252,0,8
190:    Data 74,109,0,6,102,2,214,193,81,205
191:    Data 255,130,82,71,8,7,0,0,102,26
192:    Data 74,121,0,251,128,4,74,121,0,251
193:    Data 128,4,74,121,0,251,128,4,74,121
194:    Data 0,251,128,4,84,66,190,124,0,8
195:    Data 102,0,254,246,70,250,0,232,78,117
196:    Data 58,45,0,4,74,109,0,6,102,2
197:    Data 226,77,83,69,74,68,103,40,12,109
198:    Data 0,1,0,6,103,0,254,212,12,109
199:    Data 0,2,0,6,103,0,0,8,214,193
200:    Data 214,193,96,14,74,46,2,230,103,8
201:    Data 66,46,2,230,96,0,254,182,1,18
202:    Data 103,252,74,18,106,252,78,113,74,144
203:    Data 74,144,74,144,78,249,35,51,4,68
204:    Data 38,144,214,252,0,16,38,144,214,252
205:    Data 0,16,38,144,214,252,0,16,38,144
206:    Data 214,252,0,16,38,144,214,252,0,16
207:    Data 38,144,214,252,0,16,38,144,214,252
208:    Data 0,16,38,144,214,252,0,16,38,144
209:    Data 214,252,0,16,38,144,214,252,0,16
210:    Data 38,144,214,252,0,16,38,144,214,252
211:    Data 0,16,74,109,0,6,102,4,214,193
212:    Data 214,193,81,205,255,150,82,71,74,109
213:    Data 0,6,103,2,82,71,8,7,0,0
214:    Data 102,26,74,121,0,251,128,4,74,121
```

Listing 2: Ein Beispielprogramm zur Ansteuerung der Easysoft-Routinen in GFA-Basic. Das Progrämmchen überträgt das, in den Data-Zeilen enthaltene, Maschinenprogramm in einen String, um es dann auszuführen. Als Bildformat wird das DEGAS-Format angenommen. Beide Programme sind auch auf der Diskette zu diesem Heft erhältlich.

Funktion 0:

Aufruf:

Void C: Easytizer%(0, Xw, Yw, Int, Bpl, Pos, L: Adresse)

Parameter:

Xw:
Breite des Digitaliserten Bildausschnitts, die in n/64 Bildpunkten angegeben werden muß. Minimal kann Xw eins werden, wobei nur mit einer Breite von 64 Bildpunkten digitalisiert wird. Maximalwert für Xw ist 12, wobei in einer Breite von 768 Bildpunkten digitalisiert wird.

Yw:
Anzahl der Zeilen, die ab Pos digitalisiert werden. Sofern Int gleich 0 ist (Interlace erlaubt), dürfen Yw plus Pos nicht mehr als 600 betragen, da ein Videobild nur etwa 600 sichtbare Zeilen überträgt. Ist Int jedoch größer 0, dann dürfen Yw plus Pos nicht mehr als 300 betragen.

Int:
Bestimmt den Interlace-Modus. Wenn Int = 0 ist, werden beide Halbbilder (gerade und ungerade Zeilen) des Videosignals ausgewertet. Bei Int = 1 wird jedoch nur noch ein Halbbild zur Auswertung herangezogen, wodurch das digitalisierte Bild um den Faktor 2 Vertikal gestaucht erscheint. Int = 2 ist ein Sonderfall, der nur gilt, falls Bpl = 1 (2 Bildebenen) ist. Dieser Modus ist nur zu gebrauchen für Videosignalquellen, die in beiden Halbbildern die gleiche Information übertragen, also nur 312 Zeilen haben, ist aber doppelt so schnell (80 ms) da jetzt eine komplette Bildperiode nur noch halb so groß ist.
Für Int > 1 gilt, daß das digitalisierte Bild halb so hoch erscheint, was jedoch in mittlerer Auflösung durchaus sinnvoll ist.

Bpl:
Gibt die Anzahl der Bildebenen (Graustufen) an. Zwei Werte sind zugelassen:

Bpl = 0: Wählt eine Bildebene aus (Monochrom).

Bpl = 1: Wählt zwei Bildebenen aus (4-Graustufen/mittlere Auflösung).

Pos:
Bestimmt die Zeile, ab der angefangen wird, die mit Yw spezifizierte Anzahl von Zeilen zu digitalisieren. Falls Int = 0 ist, muß dieser Wert durch zwei geteilt werden.

Adresse:
Die Adresse des Bildspeichers, in den das digitalisierte Bild übertragen wird.

Der Zweck dieser Routine (Funktion 0)
Digitalisieren eines Videosignals mit verschiedenen Optionen in einen Speicherabschnitt, welcher dem Aufbau des Atari-ST Bildschirmspeichers entspricht. Der nötige freie Speicher berechnet sich zu: Speicher in Bytes = Xw★8★Yw★(Bpl+1).
Zeitbedarf: Normal 160 mS oder 80 mS falls Int = 2 und Bpl = 1.

Funktion 1:

Aufruf:
Wert=C:Easytizer%(1)

Parameter:
Wert wird 1, falls die Schnappschuß-Taste gedrückt ist.

Zweck der Funktion 1
In Wert wird der Zustand der Schnappschuß-Taste zurückgegeben.

Funktion 2:

Aufruf:
Wert=C:Easytizer%(2)

Parameter:
Wert wird 1, wenn ein Videosignal anliegt.

Zweck der Funktion 2
Prüfen, ob ein Videosignal anliegt, da Funktion 0 solange blockiert, wie kein Videosignal anliegt.

Einige Beispiele für Funktionsaufrufe
Gesetzt den Fall, Sie wollen in mittlerer Auflösung ein Videosignal von einer 312-Zeilen-Kamera mit einer Auflösung von 640★200 direkt in den Bildspeicher digitalisieren, dann könnte das folgendermaßen geschehen:

Void C: Easytizer%(0,10,200,2,1,50,L: Xbios (&H3))

Der Ausdruck 'L:Xbios (&H3)' gibt lediglich die Adresse des logischen Bildspeichers zurück.

Oder Sie wollen in hoher Auflösung ein normales Videosignal mit 625 Zeilen direkt in den Bildspeicher digitalisieren:

Void C: Easytizer%(0,10,400,0,0,50,L: Xbios(&H3))

Oder Sie wollen nicht direkt in den Bildspeicher digitalisieren, aber mit maximaler Auflösung (768★600) und 4 Graustufen arbeiten, z. B. für Bildverarbeitungszwecke:

Void C: Easytizer%(0,12,600,0,1,3,Freispeicher%)

Wobei in Freispeicher% die Adresse eines 115200 Byte großen Speicherblocks enthalten sein muß.

In Listing 2 ist ein kleines GFA-BASIC Programm abgedruckt, welches das Digitalisieren und Sichern (Speichern im Degas-Format) eines Bildes erlaubt und gleichzeitig die Anwendung der EASYSOFT-Routinen zeigt.



Bild 2: Das Menü der über die Redaktion erhältlichen Software.

Außerdem ist eine Software über die Redaktion verfügbar, die weit über das hinausgeht, was sich in einer Zeitschrift veröffentlichen läßt. Einen kleinen Vorgeschmack bietet Bild 2, in dem das Menü dieser Software zu sehen ist.

Wir hoffen, daß Sie mit dem Easytizer viele interessante Anwendungen erschließen können.

Korrekturen zu Teil 1

Leider hatten sich im Teil 1 zum Easytizer (ST Computer Nr. 10/87 Seite 58 und 59) folgende Fehler eingeschlichen:

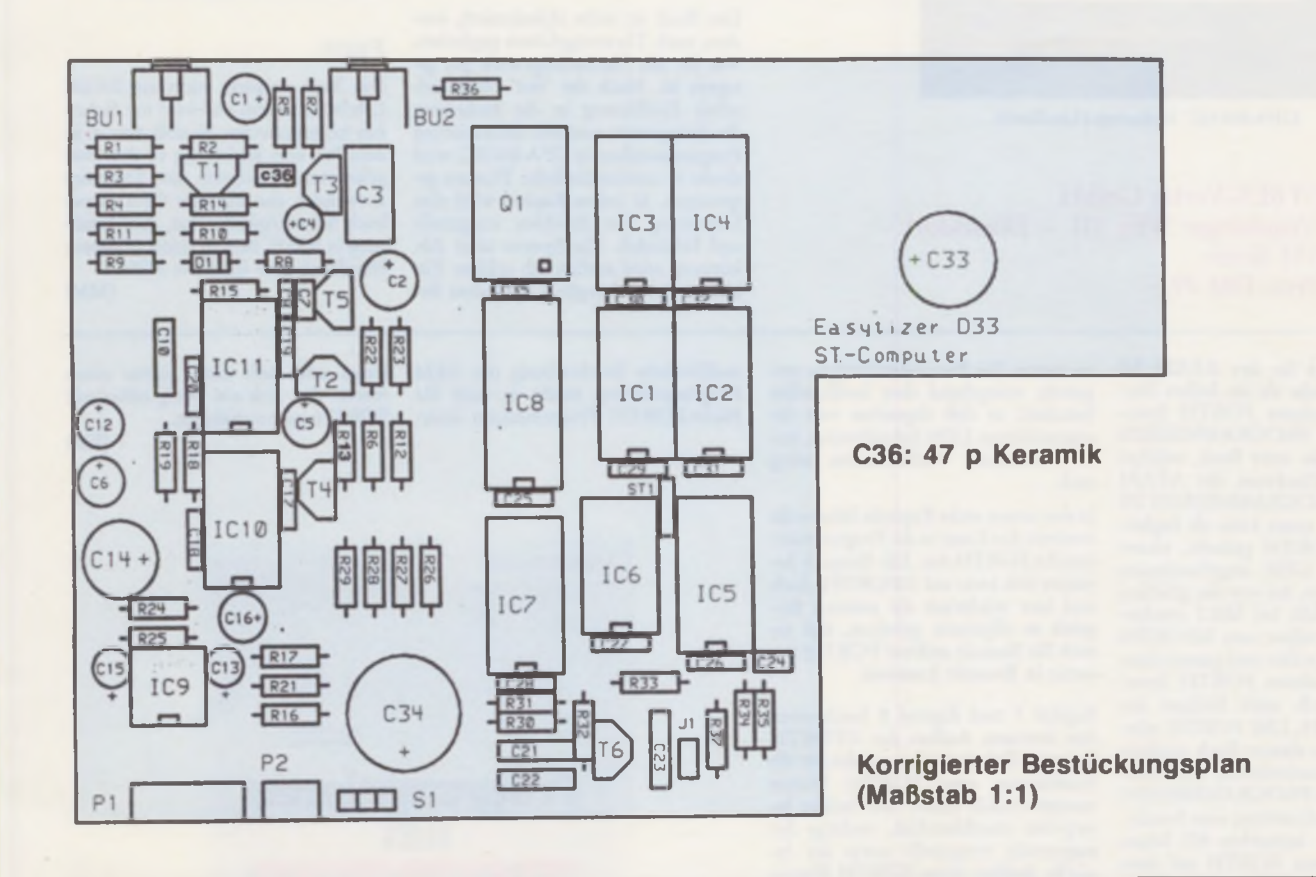
- Im Schaltbild und auf dem Bestükkungsplan war der Kondensator C35 fälschlicherweise zweimal vorhanden. Richtig ist es, den in der Videoeingangsstufe befindlichen Kondensator mit der Kapazität von 47 pF mit C36 zu benennen.
- Auf dem Bestückungsplan sind zwei Polaritätsangaben verrutscht: Bei C2 muß der Pluspol oben liegen. Bei C15 muß der Pluspol unten liegen.
- Außerdem ist R3 doppelt vorhanden. Richtig muß R3, der neben R6 liegt, in R13 umbenannt werden (siehe korrigierten Bestückungsplan).

Schließlich wurde noch verschwiegen, daß „ST1" drei Lötstifte, die für künftige Erweiterungen vorgesehen sind, vorerst nicht benötigt werden und deshalb weggelassen werden können.

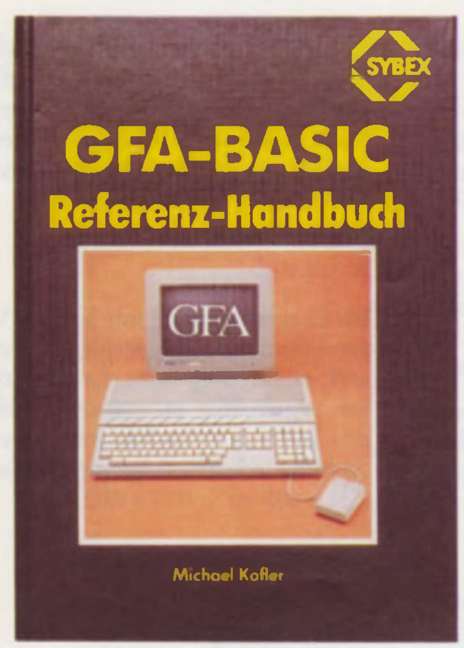
(RA)

Hinweis!

Im Zusammenhang mit einigen Videorecordern hat sich gezeigt, daß das Bild evtl. nicht ruhig steht. Grund dafür ist ein pegelmäßig zu starker Farbburst, den leider manche Recorder erzeugen. Abhilfe schafft folgende Änderung: C36 = 100 pF ; R11 = 8K2 Ohm



Korrigierter Bestückungsplan (Maßstab 1:1)



GFA-BASIC Referenz-Handbuch

SYBEX-Verlag GmbH
Vogelsanger WEg 111 – Düsseldorf
532 Seiten
Preis DM 49,–

Wie nicht anders zu erwarten, sind eine Fülle von Büchern zum Thema GFA-BASIC erschienen. Einige davon sind gut, andere dagegen weniger. Es wird aber in jedem Fall für den Anwender immer schwieriger zu entscheiden, welches Buch für ihn das 'ideale' Werk (wenn so etwas überhaupt existiert) darstellt.

Das Buch, das uns diesmal beschäftigt, ist das 'GFA-BASIC Referenz-Handbuch' von SYBEX, einem Verlag, der sich seit langem schon einen Namen durch qualitative Produkte gemacht hat. Der Autor, Michael Kofler, ist in ST-Kreisen auch schon durch sein älteres Werk, 'Das ATARI ST Grafikbuch' (im selben Verlag erschienen) bekannt.

Das Buch ist nicht alphabetisch, sondern nach Themengebieten gegliedert, was für ein Nachschlagewerk gut geeignet ist. Nach der 'fast' obligatorischen Einführung in die Bedienung des Interpreters und den theoretischen Programmaufbau in GFA-BASIC, wird direkt in unterschiedliche Themen gesprungen. In jedem Kapitel wird eine Sammlung von Befehlen vorgestellt und behandelt. Die Syntax samt Abkürzung wird ausführlich erklärt. Ein kleines Beispiel ergänzt bei jedem Befehl die meist trockene Erklärung. Am Ende jedes Befehls wird zu anderen Befehlen, die ein Komplement oder eine Verbesserung darstellen, verwiesen.

Das letzte Kapitel beschäftigt sich ausführlich mit dem Compiler. Es werden die verschiedenen Optionen zur Steuerung des Compilers, sowie eine mögliche Geschwindigkeitsoptimierung von Programmen gezeigt.

Das Buch behandelt den Interpreter in der Version 2.0 und ist insofern immer noch aktuell und interessant. Die Interpret erversion, die den mathematischen Coprozessor unterstützt, wird in dem Buch nicht besprochen, da sie erst seit kurzem im Handel ist. Als Anhang und kleine Zugabe wird ein Patch-Programm für den beschriebenen Interpreter abgelistet, so daß jeder BASIC Anwender die Fehler dieser Version selbst korrigieren kann.

Fazit

Das Buch versucht nicht ein BASIC Lehrbuch zu sein und wird mit Sicherheit keines ersetzen. Es stellt aber in jedem Fall eine Ergänzung zu dem mitgelieferten Handbuch dar. Derjenige Anwender, der mit dem GFA Handbuch Schwierigkeiten hat, aber Erfahrung in BASIC besitzt, findet in diesem Handbuch eine fundierte Hilfe.

(MM)

Zwar existieren für den ATARI ST mittlerweile mehr als ein halbes Dutzend verschiendener FORTH Systeme, doch ist PROGRAMMIEREN IN FORTH das erste Buch, welches sich auf die Hardware des ATARI ST bezieht. PROGRAMMIEREN IN FORTH ist in erster Linie als Begleitbuch zu 32FORTH gedacht, einem vollständig in GEM eingebundenem FORTH System, das von den gleichen Autoren ebenfalls bei M&T erschienen ist. Der Aufbau von 32FORTH entspricht im großen und ganzen dem der meisten anderen FORTH Systeme, so daß z.B. auch Besitzer des VOLKSFORTH, LMI FORTH oder 4xFORTH von diesem Buch profitieren werden. Zumindestens vom Umfang her nimmt PROGRAMMIEREN IN FORTH unbestritten eine Sonderstellung ein: – immerhin 531 Seiten sind dem Thema FORTH auf dem ATARI ST gewidmet. Doch keine Angst, Sie müssen das Buch nicht bis zur letzten Seite durcharbeiten, u.n den ATARI ST in FORTH programmieren zu können, denn der größte Teil des Buches befaßt sich mit der Beschreibung der AES-und VDI Routinen, die 32FORTH selbstverständlich unterstützt.

Durch die ausführliche Beschreibung der einzelnen VDI- und AES-Routinen, die durch viele Beispiele veranschaulicht werden, empfiehlt sich PROGRAMMIEREN IN FORTH als ein beinahe ideales Trainingsbuch für die VDI/AES Anwendung. Vorausgesetzt man kennt sich ein wenig in FORTH aus, lassen sich die zahlreichen Beispiele des Buches zusammen mit 32-FORTH (oder einem anderen System, das ebenfalls diese Routinen unterstützt) ohne großen Aufwand interaktiv testen. Die Parameterübergabe entspricht weitgehend dem inoffiziellen Standard, so daß abgesehen von der ungewohnten UPN Schreibweise, keine besonderen Vorkenntnisse nötig sind.

In den ersten sechs Kapiteln führen die Autoren den Leser in die Programmiersprache FORTH ein. Alle Beispiele beziehen sich zwar auf 32FORTH, doch sind hier wiederum die meisten Beispiele so allgemein gehalten, daß sie auch für Besitzer anderer FORTH Systeme in Betracht kommen.

Kapitel 7 und Kapitel 8 beschreiben den internen Aufbau des 32FORTH Systems. Dabei werden u.a. der für die Ausführung eines FORTH Wortes verantwortlich innere und äußere Interpreter durchleuchtet, wichtige Systemworte vorgestellt sowie der typische Aufbau eines FORTH Wortes erläutert.

Kapitel 9 und 10 beschreiben auf über 200 Seiten die GEM Programmierung unter FORTH. Zu den wichtigsten Funktionsaufrufen sind Beispielprogramme abgebildet, die auch auf der beiliegenden Diskette zu finden sind. Dazu gehört z.B. ein vollständig in GEM eingebundes Telefonverzeichnis, dessen Quellcode im Buch ausführlich beschrieben wird. Mit Hilfe dieser Beispielprogramme sollte der Einstieg in die GEM Programmierung eigentlich kein Problem mehr sein.

Fazit

PROGRAMMIEREN IN FORTH-ATARI ST ist ein mit viel Sorgfalt geschriebenes Buch, das zu einem außergewöhnlich niedrigen Preis eine Fülle nützlicher Informationen enthält. Die ausführliche Beschreibung der GEM Programmierung macht es auch für Nicht-FORTH Programmierer interessant und stellt nicht zuletzt einen Anreiz dar, sich ein wenig näher mit FORTH zu beschäftigen.

(PM)



Buchbesprechung: Programmieren in FORTH – ATARI ST

von Rainer Aumiller und Denis Luda – erschienen bei M&T, 1987

Die dritte Dimension

Das neue CAD 3D

CAD ist ein Zauberwort, das in zunehmendem Maße auch von der ST-Computergemeinde in den Mund genommen wird. Die Zahl der Programme ist kaum noch zählbar und ununterbrochen kommen neue Programme hinzu. Ein interessanter Neuling aus den USA ist das CAD-3D von Tom Hudson, dem Schöpfer des DEGAS Malprogramms, das als Urvater der ST-Malprogramme angesehen wird.



Computer Aided Design

bedeutet nichts anderes, als ein vom Computer unterstütztes Design, wird aber meistens mit computerunterstütztem Konstruieren übersetzt. Die meisten CAD-Programme für den ST versuchen deshalb eine Hilfe für den Maschinenbauer, den Elektrotechniker oder den Bauingenieur zu sein. Obwohl deren Objekte 3-dimensional sind, hat sich in diesen Berufsgruppen ein 2-dimensionales Zeichenverfahren durchgesetzt, weil nicht genügend Rechen- und Zeichenkapazität für eine 3-dimensionale Darstellung zur Verfügung stand. Außerdem ist es unmöglich aus einer perspektivischen Darstellung eines Objektes das entsprechende Original zu erstellen. Kann man da mehr als ein Spielzeug erwarten? Ist das Kürzel CAD im Sinne einer seriösen Konstruktionshilfe für den Techniker überhaupt angebracht?

Der Lieferumfang

Dieser Test bezieht sich auf das Cyber Studio der Firma Antic Software Publishing. Man erhält für 90$ eine ca. 100 Seiten dicke Anleitung und zwei einseitig formatierte Disketten mit den beiden Programmen Stereo CAD-3D 2.0 und Mark Kimball's CYBERMATE. Dazu kommen noch Hilfsprogramme für Hardcopy und Animation, wenn man auf die Benutzung des CYBERMATE verzichten will (oder aus Speichermangel verzichten muß!). Kern des Softwarepakets ist das CAD-3D Programm. Das CYBERMATE Programm übernimmt lediglich die Animation der mit CAD-3D erzeugten Objekte. Offensichtlich ist mir die neuste Version zuteil geworden, denn entgegen den Ankündigungen im Handbuch, lief die gesamte Software einwandfrei im Monochrom Modus. Die Farbversion ist erheblich beeindruckender. Die gedruckte Anleitung ist nur für das CAD-3D Programm gedacht. Sie enthält eine ausführliche Bedienungsanleitung, eine übersichtliche Tabelle über vorprogrammierte Tastenbelegungen und eine tabellarische Zusammenfassung aller Icons und Menüpunkte. Außerdem ist ein sehr lehrreiches Drehbuch für einen 3D-Trickfilm enthalten, das man unbedingt verfilmen sollte, wenn man die Kombination CAD-3D-CYBERMATE verstehen lernen will. Die Anweisung für das recht komplizierte und leistungsfähige CYBERMATE Animationsprogramm ist als Diskfile vorhanden (ergibt 72 Seiten Ausdruck: Neues Farbband besorgen!). Benutzer ohne eigenen Drucker können das Listing beim Hersteller bestellen (Kosten $7.50 Versand, Zoll und Mehrwertsteuer). Zum Betrieb der Programme ist dringend eine Festplatte zu empfehlen. Wer diese Programme beruflich nutzen möchte, sollte außerdem die Anschaffung eines 4 MByte ST überdenken. Bei meinen ersten Versuchen ist es mir leicht gelungen, einen über 820kByte großen ★.DLT File zu erzeugen. Dazu kommt noch der Bildfile mit 32 kByte und das Programm war nur noch mit dem Animationsprogramm, nicht aber mit dem CYBERMATE zur Animation zu bringen.

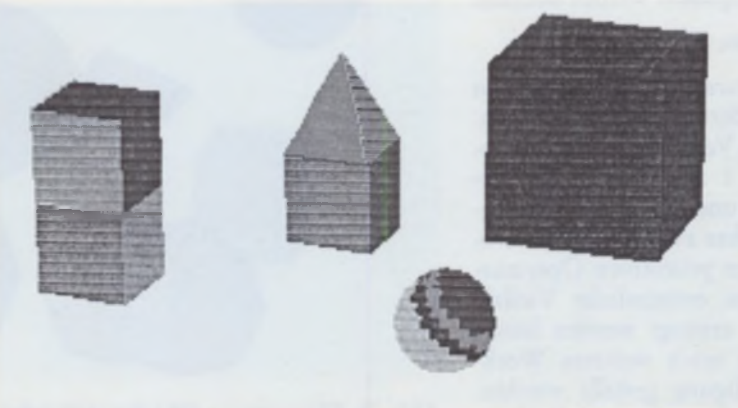


Abb. 1: Blockwelt mit CAD-3D erzeugt

Das CAD-3D Programm

Dieses Programm ist der zentrale Punkt des Softwarepaketes. Es ermöglicht nicht zunächst die Erstellung räumlicher Objekte und ihre perspektivisch korrekte Darstellung auf dem Bildschirm. Abb 1 zeigt einige einfache Körper, wie sie mit dem Programm zu erstellen sind und im sogenannten SUPERVIEW Modus angeschaut werden können. Diese Bilder lassen sich auch im DEGAS oder Neochrome Format abspeichern und weiterverarbeiten. Abb. 2 zeigt das Desktop für CAD-3D. Auf der rechten Tischseite sind die ICONS für die Objektmanipulation untergebracht, auf der rechten Hälfte sind vier Fenster zu erkennen. Das CAMERA-Fenster zeigt eine perspektivische Ansicht der Objekte. Die anderen Fenster sind TOP (BOTTOM), LEFT (RIGHT), FRONT (BACK). In Klammern stehen die Ansichten, die von der Menüleiste her umgeschaltet werden können. Übrigens sind alle Funktionen auch über die Tastatur einstellbar. Es gibt auch keine Probleme mit der deutschen Tastatur. Die deutschen Sonderzeichen sind nicht zugänglich.
Normalerweise ist nur ein Arbeitsfenster und das Kamerafenster geöffnet, um den Zeitaufwand beim Update eines Bildes zu reduzieren. Es lassen sich aber auch alle Fenster synchron zuschalten. Das jeweilige Arbeitsfenster kann auch vergrößert dargestellt werden, damit präzisere Objektmanipulationen möglich sind. Ein solches vergrößertes Fenster füllt dann die gesamte rechte Desktophälfte.

Die Erzeugung von Objekten

Elementarobjekte

Zunächst einmal stehen vier elementare Objekte zur Verfügung, die anschließend manipuliert werden können:

Kugel, Torus, Würfel und Ecke.

Diese elementaren Objekte können nun miteinander verknüpft werden. Die möglichen Verknüpfungsoperationen zeigt Tab. 1. Abb. 7 zeigt die Elementarobjekte und die möglichen Verknüpfungsobjekte zweier Kugeln. Obwohl mit diesen primitiven Operationen schon eine erstaunliche Vielfalt von Objekten erzeugt werden kann, muß natürlich noch weiteres Werkzeug zur Verfügung gestellt werden. Dieses sind der Extruder und die Spinoperation.

Tabelle 1: Verknüpfungen zweier Objekte in CAD-3D

Verknüpfung	Erklärung
Add	Das neue Objekt entsteht als Summe der einzelnen Objekte. Beispiel: Turm add Halle = Kirche. Bedingung: Beide Objekte müssen sich überlappen.
Subtract	Das zweite Objekt wird vom ersten abgezogen. Das heißt, der überlappende Teil des zweiten Objektes wird aus dem ersten entfernt.
And	Das neue Objekt ist der Raumteil, der von beiden Objekten eingenommen wird.
Stamp	Der zweite Körper wird auf die Oberfläche des ersten gestempelt. Die Fenster in Abb. 6 sind so entstanden.

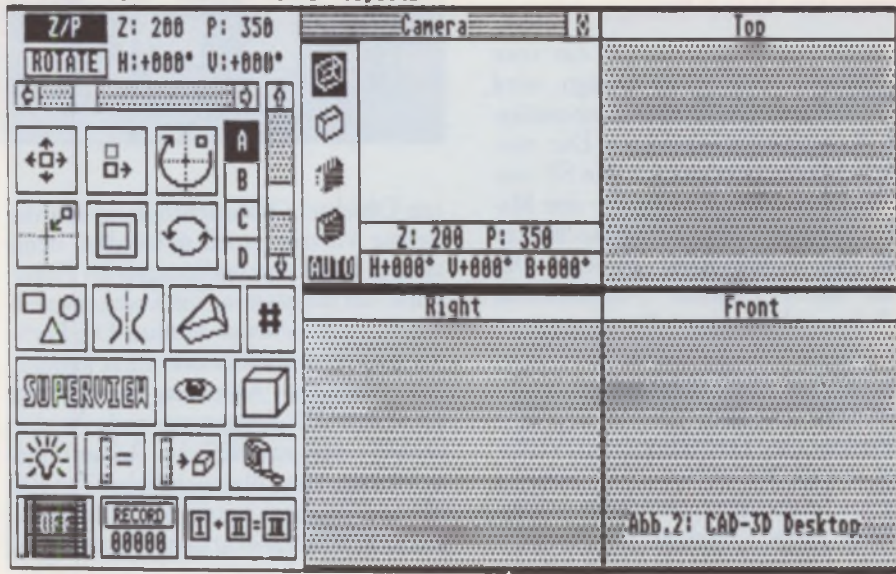


Abb. 2: CAD-3D Desktop

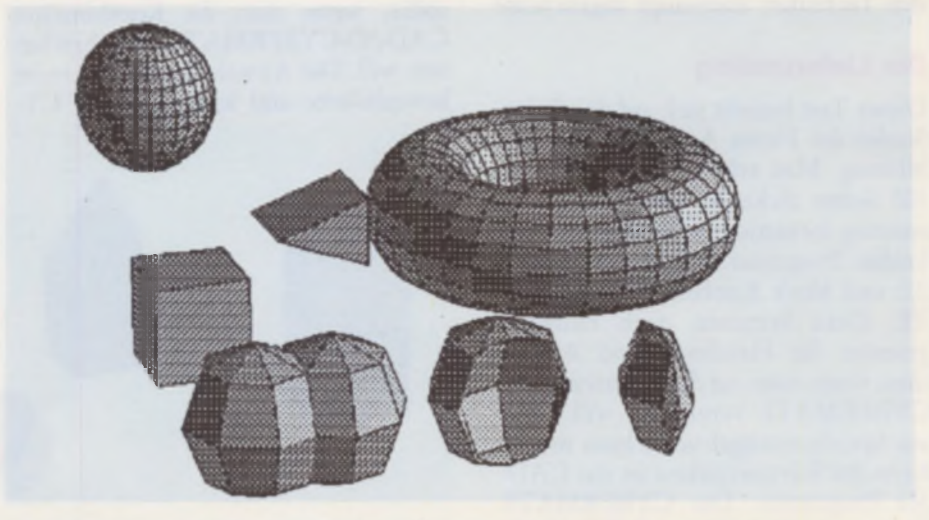


Abb. 7: Elementare Objekte (oben) und durch elementare Verknüpfung erzeugte Objekte (vorne).

Der Extruder

Vielleicht haben Sie schon einmal Laubsägearbeiten angefertigt? Das genau macht der Extruder in CAD-3D! Man gibt mit Hilfe der Maus eine Schablone ein und der Extruder genannte Programmteil sägt die Schablone aus einem imaginären Block aus. Der entstehende Block kann anschließend in alle Richtungen gedehnt und gestaucht werden. Außerdem muß der ausgesägte Block nicht massiv sein, sondern er kann aus mehereren Segmenten bestehen, die sich allerdings nicht trennen lassen (denken Sie an zusammengeleimte Holzschichten). Sowohl die erzeugten Objekte, als auch die Schablonen lassen sich natürlich speichern und bei Bedarf wieder laden!

Spin

Spin ist das Verfahren zur Erzeugung rotationssymmetrischer Körper. Auch hier ist es möglich mit Hilfe der Maus eine Schablone einzugeben. Die Schablone wird spiegelsymmetrisch zur Y-Achse abgebildet. Der Körper entsteht durch Rotation der Schablone um die Y-Achse. Alle unter Extrude beschriebenen Speicher- und Ladeoperationen sind auch unter Spin möglich.

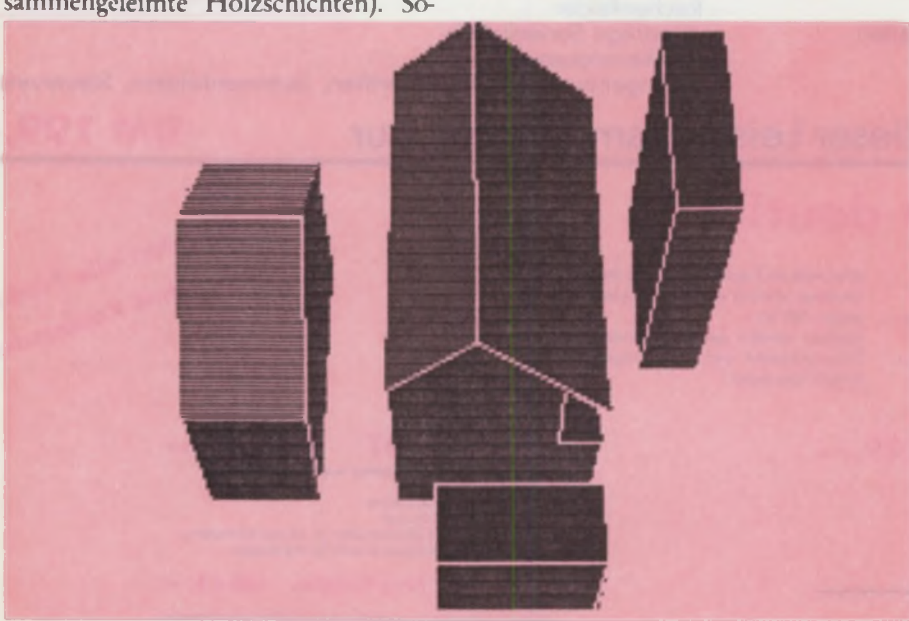


Abb. 5: Aus diesen Bestandteilen wird ein Haus

Objektmanipulationen

Es ist streng zu unterscheiden zwischen Objektmanipulationen, welche die physischen Abmessungen der Objekte verändern, und der Veränderung des Standpunktes, welcher das Objekt bei unveränderten Abmessungen unterschiedlich auf dem Bildschirm erscheinen läßt. Beschäftigen wir uns zunächst mit den Möglichkeiten, ein Objekt zu verändern. Alle Objekte können einzeln oder als Gruppe vergrößert, verkleinert oder rotiert werden. Das geschieht bei Auswahl der Gruppe (A, B, C oder D) und selektieren der entsprechenden Objekte durch Anklicken. Anklicken des Resize-Icons gibt die Möglichkeit, die gewählte Gruppe auf 50%–200% der derzeitigen Größe zu bringen. Im aktiven Fenster können die Objekte (wieder als Gruppe oder einzeln) mit Hilfe der Maus verschoben werden. Schließlich können die Objekte durch Anklicken des Rotationssymbol um −180° – +180° gedreht werden. Je nachdem in welchem Fenster die Drehung erfolgt, erhält man damit eine Drehung um die x-, y- oder z-Achse. Die Drehachse erscheint in jedem Fenster als Fixpunkt, der zudem auf verschiedene Arten festgelegt werden kann. Standardmäßig ist der Fixpunkt der Mittelpunkt des jeweiligen Fensters. Es läßt sich aber auch der Mittelpunkt der Gruppe als Fixpunkt wählen. Die letzte Möglichkeit erlaubt es, mit Hilfe der Maus einen Fixpunkt selbst zu wählen. Alle bisher geschilderten Operationen erhielten die Proportionen des Objektes. Wenn ein Körper gedehnt oder gestaucht werden soll, müssen andere Verfahren angewendet werden. Wenn das Objekt nicht maßstabsgetreu sein muß, bietet sich als einfachste Möglichkeit die Skalierung mit Hilfe der Schieber im linken Desktopteil an. Man wählt zunächst das geeignete Fenster aus, in dem die Verzerrung erfolgen soll. Dann verschiebt man den horizontalen oder vertikalen Schieber, bis er die entsprechende Dehnung oder Stauchung anzeigt und klickt den Fensterinhalt an. Alle, im aktiven Fenster befindlichen Objekte werden nun entsprechend verändert. Für technische Anwendungen ist die Einhaltung eines Maßstabes unverzichtbar. Um das zu erreichen wählt man zunächst ein Objekt aus, das als Maßstab dienen soll. Alle anderen Objekte erhalten damit automatisch die entsprechende Dimensionierung. Als Maß stehen das englische Feet oder das Meter zur Verfügung. Anschließend ist es möglich, alle Objekte maßstabsgerecht zu verändern, oder mit Hilfe des Maßbandes nachzumessen. Alle Operationen, die ein Objekt verändern, sind nur in einem der drei Fenster Top (Bottom), Right (Left) oder Front (Back) möglich, weil nur in diesen Fenstern maßstäbliches Arbeiten ohne perspektivisches Verzeichnung möglich ist. Folglich sind alle Operationen, die den Standpunkt des Betrachters bzw. die Perspektive der Kameraeinstellung betreffen, nur im Kamerafenster möglich.

Abb. 5 zeigt die Bestandteile eines Hauses. Durch die oben geschilderten Manipulationen wird daraus das in Abb. 6 gezeigte schmucke Häuschen!

Veränderung der Ansicht eines Objektes (Gruppe)

Man klickt zunächst das Kamerafenster an, damit es aktiv wird. Dann klickt man entweder das ROTATE Icon, oder das Z/P Icon an. Ist das ROTATE Icon aktiv, ist es möglich den Kamerastandpunkt auf einer Kugeloberfläche mit Hilfe der Schieber zu verschieben. Die Kamera zeigt dabei immer auf den Mittelpunkt des CAD-3D Universums. Durch die beiden Schieber ist es also möglich, das Objekt von allen Seiten zu betrachten. Der Kameraabstand wird durch Regeln des Z-wertes (Z wie ZOOM) eingestellt. Damit ist es möglich, Details eines Objektes genauer zu betrachten. Der P-wert (P wie Perspektive) gibt an, ob es sich um ein Weitwinkelobjektiv (stark stürzende Linien) oder Teleobjektiv (fast Parallelprojektion ohne stürzende Linien) handelt. Aller-

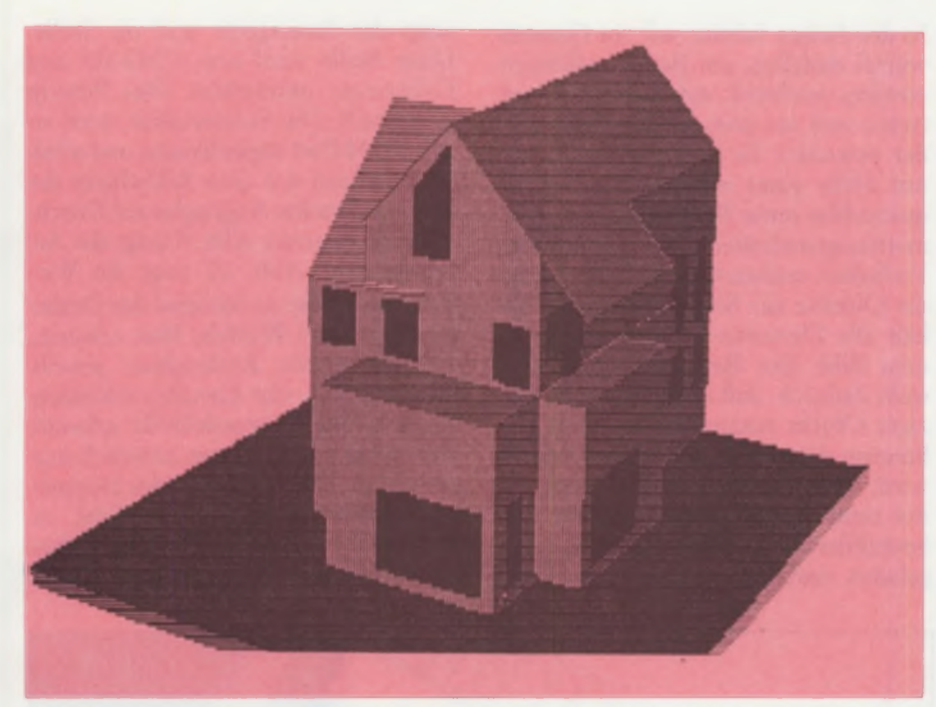


Abb. 6: Fertig zusammengesetztes Haus

dings bleibt im Gegensatz zu realen Objektiven der Bildausschnitt gleich. Hat man sein Objekt maßstabsgerecht skaliert und die Kameraeinstellung eingerichtet, dann möchte man sein Objekt natürlich betrachten. Bis jetzt nämlich haben auch Colormonitor-Benutzer lediglich schwarz-weiß Bilder gesehen. Man stellt zunächst noch die Beleuchtung ein (3 Lichtquellen können in allen Raumrichtungen verschoben werden und in Ihrer Helligkeit eingestellt werden. Dazu kommt noch eine diffuse Lichtquelle, deren Helligkeit ebenfalls eingestellt werden kann.) und klickt dann das Superview Icon an. Für Farbmonitorbesitzer kommt nun der Aha-Effekt. Nach einer kurzen Berechnungspause erscheint das strahlende Farbbild in niedriger Auflösung. Schwarz(-weiß)-seher bekommen weniger Dramatisches in hoher Auflösung zu sehen. Wenn einem das Gezeigte gefällt, sollte man zweierlei tun: Speichern und nochmals speichern. Einmal nämlich das Objekt als ★.3D2 File.

Dieses File kann später wieder eingelesen werden, egal ob gerade ein S/W- oder Colormonitor angeschlossen ist. Zum anderen als DEGAS, COLOR oder Neochrome File ★.PI?. Dabei muß die Endung nicht eingesetzt werden, CAD-3D setzt die Endung entsprechend des momentan angeschlossenen Monitors ein und erzeugt ein entsprechendes Fileformat. Das resultierende Bild kann mit den entsprechenden Malprogrammen weiterverarbeitet werden. Übrigens arbeitet CAD-3D nur im monochromen Modus oder mit mittlerer Auflösung, die Bilder werden aber entweder in hoher Auflösung oder niedriger Auflösung erzeugt.

Der ST als elektronisches Trickfilmstudio

Wie man unschwer dem Filmicon entnimmt, hat man mit dem CAD-3D gleichzeitig ein elektronisches Trickfilmstudio erworben. Wird dieses Icon angeklickt, öffnet sich eine File Selektor Box. Man gibt jetzt den Filenamen ein, unter dem die im weiteren Verlauf zu erzeugende Bildsequenz abgespeichert wird. Ersparen Sie sich die Eingabe der Endung, sie wird von CAD-3D selbständig vorgenommen. Das liegt daran, daß jetzt zwei Files erzeugt werden. Der eine erhält die Endung ★.PI1 (oder ★.PI3) und enthält das erste Superview Bild. Der zweite File erhält die Endung ★.DLT für DeLTa. Im technischen Sprachgebrauch steht Delta für Differenz. In diesem File ist angegeben, wie sich das folgende Bild vom vorhergehenden unterscheidet. Mit diese Technik ist es möglich auf einem 1 MByte Rechner möglich auf einem 1 MByte Rechner einen Film zu drehen, der nach Angaben des Herstellers normalerweise einen 10 MByte Rechner erfordert. Immerhin kann ich bestätigen, daß ein Film mit ca. 100 Bildern und einem sehr komplizierten Objekt (dem Haus in Abb. 6) ca. 800 kByte Diskettenplatz benötigt. An Bildspeicher wären für diese Sequenz ca. 3 MByte erforderlich. Ernsthafte Anwender werden dem 4 MByte Mega-ST entgegenfiebern. Und ohne Festplatte sollte man dieses System nicht für seriöse Arbeiten in Betracht ziehen. Jedenfalls kann man jetzt Manipulationen vornehmen und jedesmal nach vollzogener Manipulation auf das Record Icon klicken. Dann wird die jeweilige Änderung zum vorherigen Bild im ★.DLT-File festgehalten und der Record-Zähler zählt um eine Stelle höher. Während

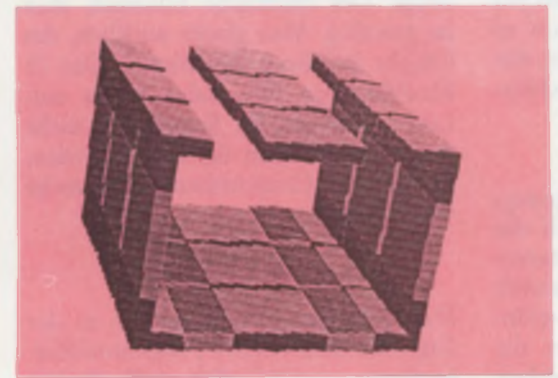


Abb. 3: Aus CUBEROOM und CUBEBALL wird ein Trickfilm

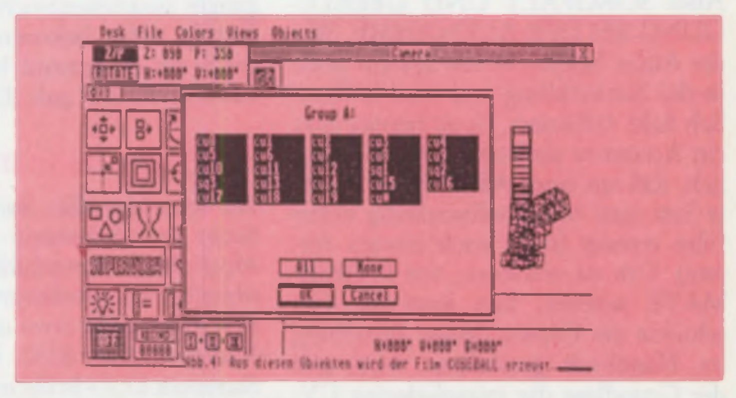


Abb. 4: Aus diesen Objekten wird der Film CUBEBALL erzeugt

dieser Aufnahme sind bestimmte Änderungen nicht möglich. Man kann während einer Aufnahme das Programm nicht verlassen, Files löschen, die Superview-Art einstellen, oder die Farbpalette ändern. Der noch freie Speicherplatz und der bisher verbrauchte Speicherplatz werden ständig angezeigt, um evtl. abbrechen zu können, falls der Speicherbedarf zu groß wird. Nochmaliges Anklicken des Record-Icons schließt die Aufnahme der Sequenz ab. Sie haben jetzt das Rohmaterial für Ihren Film „im Kasten". Nun beginnt die Schneidearbeit im CYBERMATE-Programm. In Zukunft wird es ein Accessory geben, welches diese Dreharbeit für den geplagten Kameramann übernimmt. Ich konnte mich bereits in einer Vorabversion von der Leistungsfähigkeit überzeugen.

Das CYBERMATE Programm

Das CYBERMATE Programm ist eigentlich eine Trickfilminterpretersprache. Die Syntax ist Forth-ähnlich aufgebaut. Wer sich mit dem Editor Systems aus der PD auskennt, sollte mit diesem System und der Dokumentation auf der Diskette keine Schwierigkeit haben. Trotzdem empfehle ich dringend, das mitgelieferte Beispiel zu verfilmen und anschließend im Cybermate-Programm zu bearbeiten. Man lernt dabei, das gleiche Sequenzmaterial auf verschiedenste Art und Weise zu immer neuen Filmen zusammenzusetzen. Leider besteht nicht die Möglichkeit, einzelne Bilder zu editieren. Wenn Sie also in einer 100 Bilder langen Sequenz, für deren Produktion Sie (ohne Erzeugung des Objektes) mindestens einen langen Arbeitstag ansetzen sollten, irgendwo einen Fehler machen (was Sie erst nach Abschluß der „Dreharbeiten" sehen), dann dürfen Sie schlicht alles nochmals aufnehmen. Also: SORGFALT UND ÜBERLEGUNG BEI DER AUFNAHME. Wie der Autor betont, ist das System noch in der Entwicklung und wird hoffentlich bald verbessert. Es ist bereits jetzt ein Accessory als Vorabversion erhältlich, daß aus einer ASCII-Datei Befehle liest und somit selbstständig einen Film erzeugt (siehe auch letzten Absatz). Um zu erläutern, wie CYBERMATE arbeitet, hier kurz die Geschichte des Filmes, dessen Erstellung im Handbuch geschildert wird und der Grundlage der mitgelieferten CYBERMATE-Programme ist. Abb. 3 zeigt den sogenannten CUBERROOM. In den linken Schlitz soll ein Gummiwürfel einfallen, am Boden reflektiert werden, nochmals auf den Boden auftreten und aus dem rechten Spalt wieder austreten. Es ist verblüffend, wie mit Hilfe eines relativ einfachen Sequencefiles unter Benutzung von Symmetrieeigenschaften das geschilderte Vorhaben realisiert wird. Abb. 4 zeigt die Objekte der Sequencefiles. Er enthält alle Elemente der Sequenz in einem Bild. Der Bewegungseffekt entsteht dadurch, daß immer nur ein einziges Objekt zusammen mit dem Cyberoom dargestellt und aufgenommen wird. Diese Sequenz wird zusammen mit einer bereits fertigen Sequenz und Soundfiles in das Cybermate Programm geladen und stehen dann dem Jungtrickfilmer zur Verfügung. Erst wenn man selbst gesehen hat, wie der freie Speicherplatz während des Ladevorganges zusammenbricht, wird man ein Gefühl dafür bekommen, warum es solch ein Programm bisher nicht für Microcomputer gab. Eines der schön-

STEREO – nicht 3D

Als wäre das alles noch nicht genug, bietet das Programm auch noch die Möglichkeit Stereobilder und Stereofilme(!!!) zu produzieren. Es handelt sich dabei nicht etwa um die rot-grün-Bildchen! Für $150 kann man die Stereotek-LCD-Brille mit Steuereinheit kaufen, incl. des Source Codes für eigene Anwendungen. Abb. 9 (Foto) zeigt das Steuergerät und die Brille. Diese Brille wird von CAD-3D und Cybermate unterstützt. Das Prinzip ist einfach: Die Steuereinheit wird an den ROM-Port angeschlossen und schaltet synchron mit dem Bildschirm das linke und rechte Augenglas auf Durchlaß bzw. Sperren. Abb. 8 zeigt das Arbeitsprinzip. Abb. 10 zeigt die Wirkung der Brille am Beispiel des Drahtmodelles eines Würfels. Man erkennt, daß die beiden Brillengläser jeweils nur zwei von vier Geraden erkennen lassen. Ohne Stereotek-Brille erkennt man auf dem Bildschirm natürlich nur ein mit ca. 25 Hz flackerndes Doppelbild. Dieses Prinzip ermöglicht es dann, einen Stereo Farbfilm bzw. Stereo Farbbilder zu produzieren. Natürlich sticht hier das Speicherplatzargument doppelt, denn es wird für jedes Auge ein Bild und ein Delta-File erzeugt. Der räumliche Eindruck aber ist gewaltig. Man glaubt wirklich, das Objekt berühren zu können. Das 25 Hz-Flimmern fällt dabei kaum auf. Leider können am Steuergerät nicht mehr als 2 Brillen betrieben werden, so daß eine Gruppenbenutzung nicht möglich ist.



Abb. 9: Die LCD-Stereo Brille mit Steuereinheit

G.I.S.T – Das Klanglabor

Das Cybermate Programm ist in der Lage, mit dem G.I.S.T Programm (Lieferadresse wie CAD-3D) erstellte Soundfiles synchron zum Film ablaufen zu lassen. Als Beispiel sehen Sie die in die

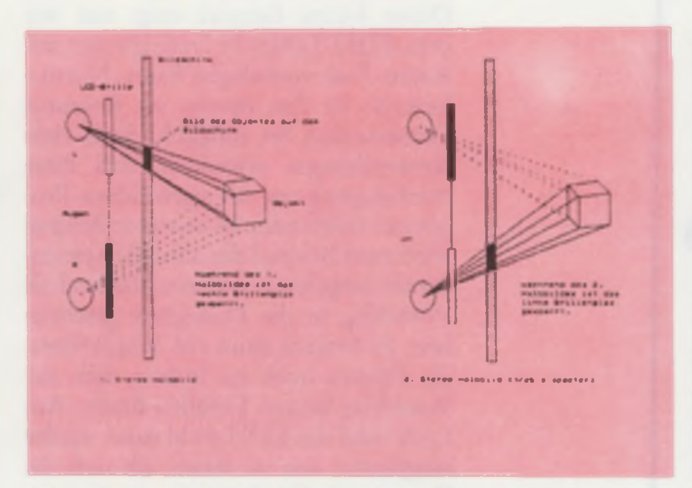


Abb. 8: Funktionsweise der LCD-Stereo Brille

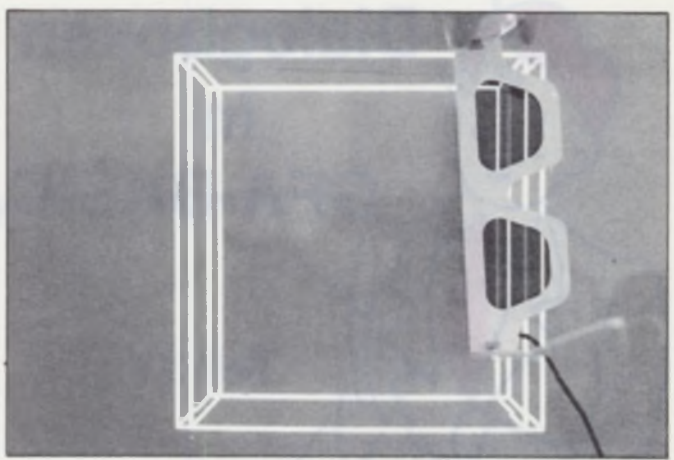


Abb. 10: Ein einfaches 3D Objekt durch die LCD Stereo Brille betrachtet. Der Effekt ist leider nicht abbildbar.

PD gegebene Filmsequenz. Man kann allerdings dieses Klanglabor auch benutzen, um seine eigenen Programme mit Klängen zu versehen. Leider ist dieses Programm nicht im Lieferumfang enthalten und muß extra bestellt werden (Preis $34.95).

Zukunftsvisionen

Daß dieses Programmpaket eine ähnliche Karriere vor sich hat, wie DEGAS vom gleichen Autor, ist anzunehmen. Bislang habe ich nichts vergleichbares von anderer Seite gesehen. Besonders erwähnenswert ist die Offenheit des Systems. Für $30 erhält man die 3D Developers Disk, die den Zugang zu CAD-3D von eigenen Programmen durch sogenannte Pipelines ermöglicht. Im Handbuch befindet sich im Anhang D das sogenannte Owners Ballot. Es handelt sich um eine Auflistung der geplanten oder angelaufenen Projekte. Diese kann der Benutzer mit einer Wertung aus seiner Sicht versehen und eigene Planungen oder Wünsche hinzufügen. Diese Karte kann der registrierte CAD-3D Benutzer dann an Antic Publishing zurücksenden. Aber schon die in dieser Karte aufgelisteten Projekte sind beeindruckend:

Advanced rendering Package:
512 Farben, geschmeidige Schattierung, Ray tracing.

Camera Motion control choreography:
Automatische Kamerasteuerung durch einen ASCII-File. Vergessen Sie was ich über die Editierung von Einzelbildern gesagt habe!

Object Motion control choregraphy:
Bewegungssteuerung von Objekten. RS-232 Eingang wird unterstützt!

Finite Element Analysis:
Vielleicht sollte man über dieses Stichwort einen eigenen Artikel schreiben?

Roland CAMM-3 driver:
Ermöglicht Computer Aided Manufacturing mit dem genannten Gerät!

Numeric input Module:
Konstruieren Sie Ihre Objekte mit Hilfe eines Wordprocessors!

Import/Export Module:
Ist etwas für Programmierer. Der Name spricht für sich!

Graftal/Fractal builder:
Erzeugen Sie synthetische Bäume, Blumen, Seen, Landschaften, Berge etc.

3D Input Device:
Eingabe eines 3D Objektes mittels Ultraschall. Einfach hinstellen, Knopf drücken und das Objekt ist im Speicher!

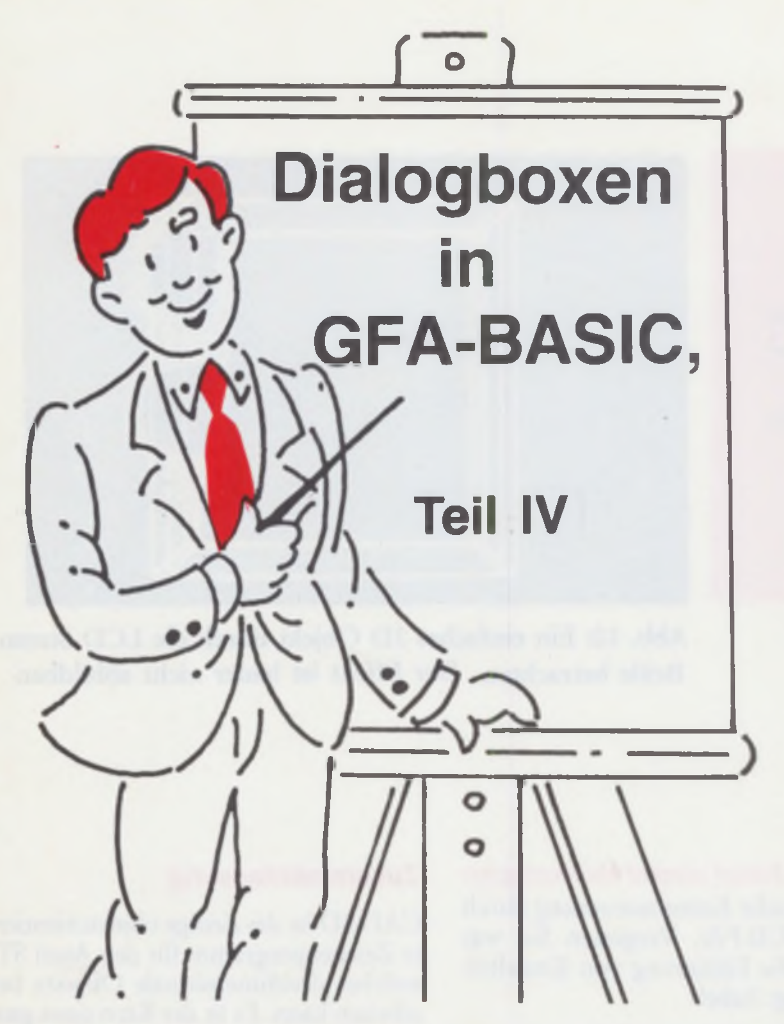
Gewiß, vieles wird noch auf sich warten lassen, aber wenn nur einige der angekündigten Projekte in näherer Zeit beendet werden, wäre das meiner Meinung nach großartig.

Zusammenfassung

CAD-3D ist das einzige objektorientierte Zeichenprogramm für den Atari ST, welches dreidimensionale Objekte bearbeiten kann. Es ist der Kern eines ganzen 3D-Computerstudios. Die Hauptzielrichtung in der gegenwärtigen Ausbaustufe, sind wohl eher die PR-Abteilungen von Betrieben. Einfache technische Anwendungen sind aber schon möglich bzw. werden (siehe obige Projektliste) geplant. Alle Objekte können jetzt schon bemaßt werden, es fehlt aber leider (noch) eine Schnittstelle für andere (zweidimensionale) CAD-Systeme bzw. für externe Datenbanken etc. Dieser Zugang ist möglicherweise mit Hilfe der 3D Developers Disk gegeben (Auch hier muß ich Sie auf einen späteren Test vertrösten!).

Meine Beurteilung: Alle, die eines ihrer Interessengebiete in der vorhergegangenen Beschreibung entdecken konnten, sollten sich dieses Programm beschaffen. Der Preis ist jedenfalls für das Gebotene mehr als fair.

Bezugsquelle
The Catalog
544 Second Street
San Francisco, CA 94107
USA
Preis: $89,95 + $10 Luftfracht.
inzwischen hoffentlich auch bei Fachhändlern
Voraussetzung: ST mit 1MByte RAM und TOS im ROM.



Dialogboxen in GFA-BASIC, Teil IV

Nachdem wir nun eine Ausgabe ausgesetzt haben, wollen wir im nunmehr letzten Teil unserer Serie noch einmal zwei kleine Beispiele zur Programmierung von Dialogboxen bringen. Mit den Routinen, die in den somit vier Teilen dieses Grundlagenkurses vorgestellt wurden, sollte es nun jedem möglich sein, selbsterstellte Dialogboxen zu verwalten.

Zuerst wollen wir ein letztes Mal eine notwendige Prozedur beschreiben, die unsere Sammlung an Routinen vervollständigen sollen.

Eine neue Prozedur

Put_dialogtext – Schreiben eines Textstrings in ein EDIT-Feld

Put_dialogtext(Baumadresse%, Objektindex%, Text$, ★Textadresse%)

Mittels dieser Routine läßt sich ein beliebiger Textstring in ein EDIT-Feld schreiben. Sie bietet somit das genaue Gegenstück zur bereits vorgestellten Routine **Get_dialogtext**. Natürlich muß man darauf achten, ob irgendwelche Einschränkungen bei den Zeichen des EDIT-Feldes bestehen, also zum Beispiel, daß nur Zahlen oder Großbuchstaben beim Erstellen der Dialogbox zugelassen werden. Die Routine kann auch anstatt der Routine **Ptadr** verwendet werden, da sie ebenfalls die Textadresse im Speicher zurückgibt und es wenig Sinn hätte, den Cursor an den Anfang des EDIT-Feldes zu setzen (siehe Teil 1 in der Juni-Ausgabe).

Als Parameter benötigen wir die Baumadresse, den Objektindex des EDIT-Feldes, sowie den eigentlichen Textstring, der in der Dialogbox erscheinen soll. Zurückgegeben wird dann nach der Abarbeitung der Routine die Textadresse in der Objektstruktur.

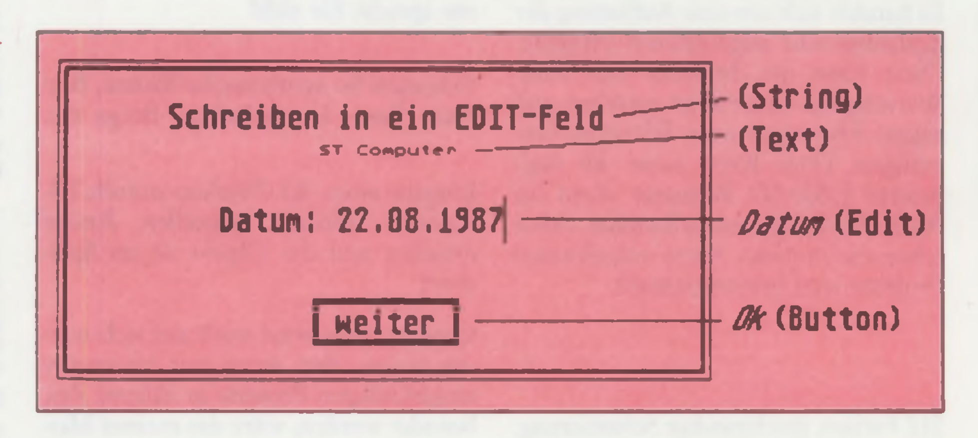


Bild 1:

Schreiben in ein EDIT-Feld

Dieses kleine Beispiel zeigt uns wie man EDIT-Felder in Dialogboxen mit einem Text vorbelegen kann. Normalerweise ist dies bereits im Resource Construction Set möglich, doch viele Anwendungen erfordern, daß diese Vorbelegung erst im eigentlichen Programm stattfindet. In unserem Beispiel wird zum Beispiel das aktuelle Systemdatum ausgelesen und anschließend als Vorschlag in die Dialogbox geschrieben. Es besteht dann die Möglichkeit, das Datum noch zu ändern, falls der Vorschlag keinen Gefallen findet. Am Ende wird das EDIT-Feld dann wieder ausgelesen, um zu sehen, ob sich das Datum geändert hat.

Im ersten Programmbeispiel werden folgende Routinen benutzt: Rsrc_load, Rsrc_gaddr, Box_draw, Form_do, Form_center, Form_dial, Objc_draw, Rsrc_free, Box_undraw, Put_dialogtext, Get_dialogtext, Undo_objc.

Numero Zwo

Das zweite Beispiel erfordert keine neuen Prozeduren, da nur alte Bekannte verwendet wurden, die schon ausreichend in vorherigen Folgen besprochen wurden. Gezeigt werden soll die Manipulierung des Objektstatusflags DISABLED und wie man leicht eine Zählbox erstellen kann, die per Maus gesteuert wird.

Zur Konstruktion

Zunächst benötigen wir in unserer Dialogbox die Objektelemente **Boxchar, G_Box** und **String** (siehe Bild 2). Wichtig ist das die beiden Boxchar-Objekte als TOUCHEXIT erklärt werden, damit die Form_do-Routine unterbrochen werden kann, wenn man

auf sie klickt und somit die Möglichkeit besteht, den String zu verändern bzw. um zählen zu können.

Ferner ist es wichtig zu wissen, daß wir die Buttons *Hundert* und *Tausend* bereits im RCS als DISABLED erklärt haben. Dies kann man natürlich auch vom Programm aus machen und soll hier nur als Erklärung dienen.

Die ganze Box funktioniert nun folgendermaßen. Man klickt einfach auf eines der beiden Boxchar-Objekte (< oder >) und sofort wird die Zahl in dem Feld zwischen ihnen herauf- oder heruntergezählt. Je nachdem welchen Wert die Zahl gerade hat, werden die Buttons durch Setzen oder Löschen des DISABLED-Flags dis- oder reaktiviert. Zahlen werden im Beispiel nur bis zu drei Ziffern (max. 999) zugelassen. Außerdem sind nur positive Zahlen erlaubt. Die Beschränkungen gelten natürlich nur für unser Beispiel. Wer gerne negative Zahlen benutzt, kann dies selbstverständlich in eigenen Programmen tun, muß aber auf die Stellenzahl des Strings achten.

In diesem Programmbeispiel werden folgende Routinen benutzt: Rsrc__load, Rsrc__gaddr, Box__draw, Form__do, Form__center, Form__dial, Objc__draw, Rsrc__free, Box__undraw, Put__dialogtext, Get__dialogtext, Do__objc, Set__string, Objc__update, Undo__objc, Objc__offset, Get__objc__xywh.

Nach den üblichen Rsrc-Prozeduren erfolgt ein normales Box__draw mit Bildschirmhintergrund retten. Danach muß das Retten des Hintergrunds aber sofort wieder abgeschaltet werden (Rette=0), da ansonsten bei jedem Aufruf von Objc__draw ein Retten erfolgen würde. Objc__draw wird ja auch von anderen Prozeduren wie z. B. Objc__update benutzt.

Die Hauptroutine läuft in einer Do... Loop-Schleife und verläßt diese nur, wenn der weiter-Button angeklickt wurde. In der Schleife werden andauernd die beiden Boxchar-Objekte abgefragt. Wird eines angeklickt, erhöht bzw. erniedrigt sich der Zähler, der neue Wert wird in den String der Zählbox geschrieben und die Zählbox mit Objc__update neu gezeichnet. Gleichzeitig wird der Wert abgefragt und die drei Buttons (0–10, 11–100, 101–1000) werden entsprechend aktiviert bzw. disaktiviert. Damit die Abfrage sowohl beim Erhöhen als auch beim Er-

```
' ********************************
' *                              *
' *   Dialogboxen in GFA-BASIC   *
' *                              *
' *       Teil IV     HE         *
' *                              *
' *  Schreiben in ein EDIT-Feld  *
' *                              *
' ********************************
'
Baum=0  !   TREE
Ok=4    !   OBJECT in TREE #0
Datum=3 !   OBJECT in TREE #0
'
Gosub Rsrc_load("nov871.rsc",*Fehler%)
If Fehler%=0 Then
  Print "RSC-Ladefehler"
  End
Endif
'
Gtype%=0
Gindex%=Baum
Gosub Rsrc_gaddr(Gtype%,Gindex%,*Baum_adr)
'
Datum$=Left$(Date$,2)+Mid$(Date$,4,2)+Right$(Date$,4)
Gosub Put_dialogtext(Baum_adr,Datum,Datum$,*Txt_adr)
Gosub Box_draw(Baum_adr,0,0,0,0)
Gosub Form_do(Baum_adr,Datum,*Ex_obj%)
Gosub Box_undraw(Baum_adr,0,0,0,0)
Gosub Undo_objc(Baum_adr,Ex_obj%,1)
Gosub Get_dialogtext(Txt_adr,*Txt$)
Txt$=Left$(Txt$,2)+"."+Mid$(Txt$,3,2)+"."+Right$(Txt$,4)
Alert 1,"Der Text lautet :| |"+Txt$,1,"weiter",A
Gosub Rsrc_free
End
Procedure Put_dialogtext(Tree%,Obj_index%,Txt$,Txt_adresse)
  '
  Te_ptext=0
  Obj_adresse%=Tree%+24*Obj_index%
  Tedinfo_adresse%=Lpeek(Obj_adresse%+12)
  Txt_adresse%=Lpeek(Tedinfo_adresse%+Te_ptext)
  I=0
  While I<Len(Txt$)
    Poke Txt_adresse%+I,Asc(Mid$(Txt$,I+1,1))
    Inc I
  Wend
  *Txt_adresse=Txt_adresse%
Return
```

Listing 1: Schreiben in ein Edit-Feld

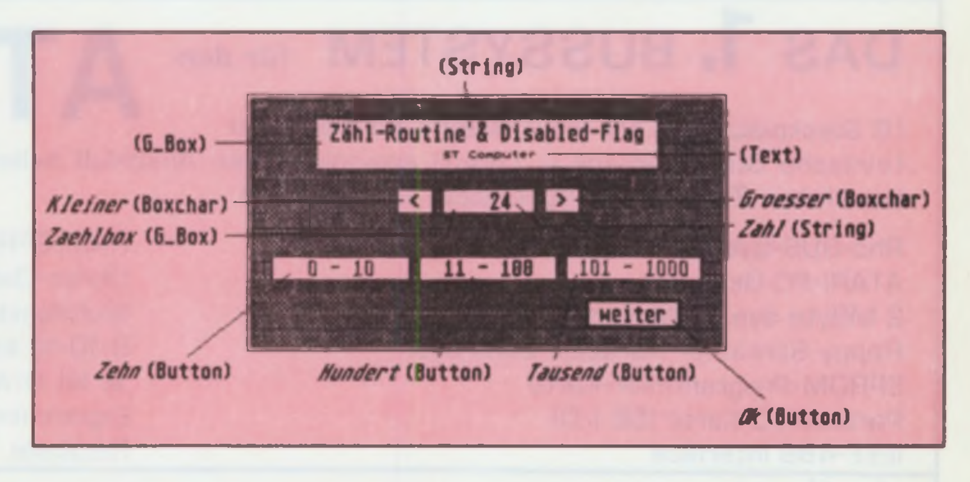


Bild 2:

```
' ********************************
' *                              *
' *   Dialogboxen in GFA-BASIC   *
' *                              *
' * Teil IV  (Zählen & Disabled) *
' *                           HE *
' *                              *
' ********************************
'
Baum=0       !   TREE
Kleiner=4    !   OBJECT in TREE #0
Zaehlbox=5   !   OBJECT in TREE #0
Zahl=6       !   OBJECT in TREE #0
Groesser=7   !   OBJECT in TREE #0
Zehn=8       !   OBJECT in TREE #0
Hundert=9    !   OBJECT in TREE #0
Tausend=10   !   OBJECT in TREE #0
Ok=11        !   OBJECT in TREE #0
'
Gosub Rsrc_load("nov872.rsc",*Fehler%)
If Fehler%=0 Then
  Print "RSC-Ladefehler"
  End
Endif
'
Gtype%=0
Gindex%=Baum
Gosub Rsrc_gaddr(Gtype%,Gindex%,*Baum_adr)
'
Rette=1
Gosub Box_draw(Baum_adr,0,0,0,0)
Rette=0
Do
  Exit If Ex_obj%=Ok
  Gosub Form_do(Baum_adr,0,*Ex_obj%)
  Zaehler_alt=Zaehler
  If Ex_obj%=Kleiner And Zaehler>0 Then
    Dec Zaehler
  Endif
  If Ex_obj%=Groesser And Zaehler<1000 Then
    Inc Zaehler
  Endif
  If Zaehler<11 And Zaehler<1000 And Disable=0 Then
    Gosub Undo_objc(Baum_adr,Zehn,8)
    Gosub Do_objc(Baum_adr,Hundert,8)
    Gosub Do_objc(Baum_adr,Tausend,8)
    Gosub Disable_update
    Disable=1
  Endif
  If Zaehler<101 And Zaehler>10 And Disable=1 Then
    If Disable=1 And Zaehler_alt<101 Then
      Disable=0
    Else
      Disable=0
    Endif
    Gosub Do_objc(Baum_adr,Zehn,8)
    Gosub Undo_objc(Baum_adr,Hundert,8)
    Gosub Do_objc(Baum_adr,Tausend,8)
    Gosub Disable_update
  Endif
  If Zaehler<1001 And Zaehler>100 And Disable=0 Then
    Gosub Do_objc(Baum_adr,Zehn,8)
    Gosub Do_objc(Baum_adr,Hundert,8)
    Gosub Undo_objc(Baum_adr,Tausend,8)
    Gosub Disable_update
    Disable=1
  Endif
  Zaehler$=Str$(Zaehler)
  Gosub Set_string(Baum_adr,Zahl,Zaehler$,4)
  Gosub Objc_update(Baum_adr,Zaehlbox)
Loop
Gosub Box_undraw(Baum_adr,0,0,0,0)
Gosub Undo_objc(Baum_adr,Ex_obj%,1)
Gosub Rsrc_free
End
Procedure Disable_update
  '
  Gosub Objc_update(Baum_adr,Zehn)
  Gosub Objc_update(Baum_adr,Hundert)
  Gosub Objc_update(Baum_adr,Tausend)
Return
```

Listing 2: Zählen und Disabled

niedrigen funktioniert, wurde das Hilfsflag **Disable** eingeführt. Da jedesmal alle drei Buttons neu gezeichnet werden müssen, haben wir dies in einer Prozedur am Ende des Programms erledigt, die jedesmal aufgerufen wird.

So, nun sind wir am Ende unseres Kurses angekommen. Hoffentlich haben wir nicht zu viele ST Leser gelangweilt, so daß nur ein lautes Aufatmen zu hören ist. Sie können uns ja einmal schreiben, was Ihnen an dieser Serie ge- bzw. mißfallen hat. In diesem Sinne wünschen wir Ihnen viel Spaß beim weiteren Programmieren!

(HE)

520 STM auf dem NEUESTEN STAND

DAS PC-GEHÄUSE



KOMPAKT-KIT BEINHALTET:

- ⋆ *Flaches, abgesetztes TASTATUR-GEHÄUSE mit RESETKNOPF und voll entstörter Schnittstellenplatine und SPIRALKABEL.*
- ⋆ *Hauptgehäuse ist vorbereitet für bis zu 2 LAUFWERKE UND EINE HARDDISK (Atari und die meisten Fremdhersteller) mit allen dazu benötigten Kabeln, Befestigungen und Blenden.*
- ⋆ *SCHALTNETZTEIL (VDE- und Post-zugelassen: versorgt Rechner, Harddisk und Laufwerke. ZENTRALER NETZSCHALTER an der Vorderseite des Hauptgehäuses.*

ALS BAUSATZ

KOMPAKT-KIT .... 398,00 DM
MIT NEC 1036A .... 598,00 DM

- ⋆ *Hauptgehäuse wird auf ST-Untergehäuse mit Zwischendeck aufgebaut, so daß ALLE URSPRÜNGLICHEN SCHNITTSTELLEN BLEIBEN. Der komplette Einbau OHNE LÖTEN – AUSFÜHRLICHE GEBRAUCHSANLEITUNG.*

ANSCHLUSSFERTIG 1298,00 DM

FERTIG UMGEBAUTER 520STM MIT 1 NEC DOPPELSEITIG. LAUFWERK, MAUS & BASIC.

HARDDISK: ZWEITES LAUFWERK: SPEICHERERWEITERUNG, ENTSPRECHENDER AUFPREIS

BALD: 1040 KOMPAKT-KIT!!!

HARDDISK-ERWEITERUNGS KIT .... 98,00 DM

- ⋆ *Benötigtes Kabel und Einbaumaterial für Atari Harddisk. (204)*
- ⋆ *Zeitverzögerungsschaltung: Gewährleistet gemeinsames Anschalten von Harddisk und Rechner über zentralen Netzschalter*
- ⋆ *Akku-Pufferung für die Uhr innerhalb des Tastaturprozessors (Akkus extra).*

SCHALTNETZTEILE ab 118,00 DM
AZTEK (VDI- & POST-ZULASSUNG)

LAUFWERKE .... 238,00 DM
NEC 1036A 3,5 DOPPELSEITIG 1 MBYTE

TASTATURGEHÄUSE . 128,00 DM

- ⋆ Flaches, abgesetztes TASTATUR-GEHÄUSE mit RESETKNOPF und voll entstörter Schnittstellenplatine und SPIRAL-KABEL.

20, 40 & 60 MBYTE HARDDISKS & STREAMERS: MONITOREN & SPEICHERERWEITERUNGEN!!

DISKETTENSTATIONEN
NEC 1036A 3,5" DOPPELSEITIG 1 MB in Gehäuse mit Stromversorgung. Voll ATARI kompatibel, Anschlussfertig.

EINZELSTATION ..... 348,00 DM
DOPPELSTATION..... 648,00 DM

SPEZIELL FÜR IHREN ATARI 260/520 ST

KOMPAKT-KIT+

BAUSATZ 398,00 DM
MIT FERTIG EINGEBAUTEM 520STM, 1 DOPPELSEITIG. LAUFWERK, MAUS & BASIC.
ANSCHLUSSFERTIG!
NUR 1.298,00 DM

Riedstr. 2 · 7100 Heilbronn **Fa. ANTHONY SEXTON**
TEL. 0 71 31/7 84 80 **Technische Entwicklung**

A & G SEXTON G.M.B.H. (i.G.)
Riedstr. 2 · 7100 Heilbronn · Tel. 0 71 31 / 784 80

Software zum fairen Preis

Jetzt oder nie:

GFA-BASIC Interpreter V2.0 **DM 88,–**

GFA-BASIC Compiler **DM 88,–**

GFA-DRAFT plus DM 299,–

Bei Vorkasse (Euroscheck beilegen) keine Versandkosten!

Bei Nachnahme zuzügl. 5,– DM

Bei Versand ins Ausland (nur gegen Euroscheck) zuzügl. 10,– DM

Weitere günstige Angebote:

GFA-MOVIE .... DM 129,–
GFA-PUBLISHER .... DM 359,–
ST-DIGI-DRUM .... DM 65,–
monoSTar .... DM 89,–
Signum! .... DM 379,–
STAD .... DM 158,–
Art-Director .... DM 179,–
OMIKRON.BASIC Disk. .... DM 159,–
OMIKRON.BASIC Comp. .... DM 147,–

Oder Bücher:

GFA-BASIC Programmier. .... DM 49,–
GFA-Handbuch TOS & GEM .... DM 49,–
GFA-BASIC-Buch .... DM 79,–

Weitere Software: Preisliste anfordern (frankierten Rückumschlag beilegen)!

RIA BEERS

Software-Vertrieb
Goddardstraße 42, 5300 Bonn 1

ST-Ecke

Bilder kopieren leicht gemacht

So manche ST-Besitzer, besonders diejenigen unter ihnen, die sich viel mit Grafik beschäftigen, sind bestimmt schon einmal auf die Idee gekommen, ein paar Ausschnitte auf dem Bildschirm zu kopieren. Nichts ist einfacher als das, denn GEM liefert uns dafür ja eine geeignete Routine, was eine GRECT-Struktur oder ein FDB ist wird im folgenden untersucht.

Ich habe die ST-Ecke dieses Monats der VDI-Routine **vro_cpyfm()** und deren Anhängsel gewidmet. Als erstes schauen wir uns die Routine mit ihren Parametern an (siehe Abb. 1):

vro_cpyfm(vdi_handle, modus, punkte, ausgang, ziel);

int vdi_handle; Kennungsnummer des VDI für die Applikation

int modus; Bitverknüpfung zwischen Ausgang und Ziel. siehe unten

int punkte[8]; Feld, das die Koordinaten von Ausgang und Ziel enthält, dabei werden die einzelnen Elemente folgendermaßen besetzt:

punkte[0]: X-Koordinate des Ausgangsbildbereiches

punkte[1]: Y-Koordinate des Ausgangsbildbereiches

punkte[2]: Breite des Ausgangsbildbereichs

punkte[3]: Höhe des Ausgangsbildbereiches

punkte[4]: X-Koordinate des kopierten Ausschnitts

punkte[5]: Y-Koordinate des kopierten Ausschnitts

punkte[6]: Breite des kopierten Ausschnitts

punkte[7]: Höhe des kopierten Ausschnittes

FDB ausgang,ziel: Form Definition Block – Struktur, die die Art des Ausschnittes kennzeichnet. In manchen Compilern wird diese Struktur auch MFDB (Memory Form Definition Block) genannt.

Abb. 1: Parameter der vro_cpyfm-Routine

Was ist ein FDB? Ein FDB ist ein sogenannter Form Definition Block, der die Art des Grafikspeichers beschreibt, aus dem oder in den der Ausschnitt kopiert werden soll. Die Bildspeicherorganisation des Atari ST ist je nach Auflösung unterschiedlich. Dabei sind folgende Kriterien vorhanden, die die Bildschirmspeicherarten unterscheiden:

a) Breite (in Pixeln)
b) Breite (in Worten – 16 Bit)
c) Höhe (in Pixeln)
d) Anzahl der Farbebenen (Bitbreite der Anzahl der Farben)

An einem Beispiel veranschaulicht: In der Monochrom-Auflösung besitzt der ST 640 (a) auf 400 (c) Punkte in 2 Farben. Um zwei Farben darzustellen, braucht man nur ein Bit, also hat die monochrome Auflösung nur eine Farbebene (Plane). Die Anzahl der 16 Bit-Worte, die in eine Zeile passen, beträgt 40. Die mittlere Auflösung hat 640 (a) auf 200 (c) Punkte in 4 (2 Planes) Farben und eine sich daraus ergebende Wortanzahl von 40. Die niedrigste Auflösung hat bei einer Anzahl von 16 (4 Planes) die meisten Farben, aber nur 320 (a) auf 200 (c) Punkte, dies entspricht einer Breite von 20 Worten. Demnach kann man an diesen Parametern sehr gut die Art (Form) des Bildspeichers beschreiben. Hier nochmal eine Übersicht der Werte:

Auflösung	Breite (Pixel)	Breite (Worte)	Höhe	Planes
monochrom	640	40	400	1
farb, mittel	640	40	200	2
farb, gering	320	20	200	4

Das einzige, was noch fehlt, ist die Adresse dieses Bildschirmbereichs. Nachdem wir diese verschiedenen Kriterien betrachtet haben, wollen wir uns den FDB anschauen, dessen Struktur hier noch einmal dargestellt ist (siehe Abb. 2).

Für uns sind nur die mit den Pfeilen gekennzeichneten Einträge interessant, da die anderen reserviert und damit vorbestimmt sind. Die mit den Pfeilen gekennzeichneten Einträge entsprechen den oben besprochenen Kriterien. Nun ein letztes Mal zur Verdeutlichung die FDBs der Bildschirmspeicher bei verschiedenen Auflösungen, wobei die Adressen natürlich nicht eingetragen sind:

```
FDB hoch =
  { 0L, 640, 400, 40, 0, 1, 0, 0, 0);
FDB mittel =
  { 0L, 640, 200, 40, 0, 2, 0, 0, 0);
FDB niedrig =
  { 0L, 320, 200, 20, 0, 4, 0, 0, 0);
```

Beachten Sie bitte das 'L' hinter der Null, das die Null als Langwort kennzeichnet. Natürlich können Sie sich einen Block in beliebiger Größe definieren. Dies ist besonders sinnvoll, wenn Sie Ausschnitte aus dem Bildschirm in den Speicher kopieren wollen. Man definiert sich einen Speicherbereich in der Größe des Ausschnittes, allerdings mit gleicher Anzahl der Planes wie des Bildschirm-FDBs. Wollen Sie einen Ausschnitt von der Größe 133 x 120 an der Stelle X=10 und Y=40 in der

mittleren Grafikstufe aus dem Bildschirm an die Stelle X=300 und Y=50 kopieren, müssen zwei FDBs erstellt werden. Wenn Sie die oben angesprochene VDI-Routine vro__cpyfm() benutzen, so entfällt die Arbeit den FDB des Bildschirms anzulegen, denn wenn Sie in die Adresse des FDBs eine 0 schreiben, wird diese automatisch vom VDI durch die aktuelle Adresse des Bildschirms ersetzt. Nicht nur das – sogar alle anderen Parameter, wie die Anzahl der Planes, etc. werden automatisch gesetzt. Wir müssen uns also in diesem Fall nicht um die FDBs kümmern, außer daß wir die Adresse auf Null setzen. Jetzt fehlt uns noch die Möglichkeit, die Koordinaten des Ausschnittes anzugeben – dies geschieht in dem Punkte-Feld, wobei das Feld für unser Beispiel folgendermaßen gesetzt wird (siehe Abb. 3).

Vorausgesetzt Sie haben VDI schon vorher initialisiert, rufen Sie die Routine folgendermaßen auf:

vro__cpyfm(vdi__handle, 3, punkte, ausgang, ziel);

wobei die beiden FDBs 'ausgang' und 'ziel' heißen, und der MODUS, um den wir uns am Ende noch genauer kümmern werden, auf 'Ersetzen' eingestellt ist. Nun wollen wir uns dem interessanteren Beispiel, dem Speichern eines Ausschnittes im Speicher des Rechners zuwenden. Dabei bleiben der Ausgangs-FDB wie auch die ersten vier Feldelemente des Feldes „punkte[]" wie im vorherigen Beispiel bestehen. Den zweiten FDB definieren wir uns genauso groß wie der Ausschnitt ist. Das bedeutet, daß die folgenden Einträge des Ziel-FDBs folgendermaßen aussehen müssen:

```
ziel.fd__w = 133;
ziel. fd__h = 120;
```

Die Wortbreite müssen wir uns errechnen, wobei ein angefangenes Wort (die Breite läßt sich in den wenigsten Fällen glatt durch sechzehn teilen) natürlich mitzählt:

```
ziel.fd__wdwidth = (ziel.fd__w)/16 +
((ziel.fd__w)%16 != 0);
```

Die Elemente fd__stand und fd__nplanes werden mit den gleichen Werten versehen, wie sie auch im Ausgangs-FDB vorhanden sind. In unserem Beispiel:

```
ziel.fd__stand = 0;
ziel.fd__nplanes = 2;
```

```
long fd__addr;      /* <- Adresse */
int fd__w;          /* <- Breite in Pixel */
int fd__h;          /* <- Höhe in Pixel */
int fd__wdwidth;    /* <- Breite in Worten */
int fd__stand;      /* Standardformat 1=Ja 0=Nein */

int fd__nplanes;    /* <- Anzahl der Planes */
int fd__r1;         /* reserviert 0 */
int fd__r2;         /* reserviert 0 */
int fd__r3;         /* reserviert 0 */
```

Abb. 2: FDB-Struktur

```
punkte[0] = 10;                  /* X-Koordinate Ausgang */
punkte[1] = 40;                  /* Y-Koordinate Ausgang */
punkte[2] = punkte[6] = 133;     /* Breite */
punkte[3] = punkte[7] = 120;     /* Höhe */
punkte[4] = 300;                 /* X-Koordinate Ziel */
punkte[5] =  50;                 /* Y-Koordinate Ziel */
```

Abb. 3

Den zweiten Teil des punkte[]-Feldes belegen wir folgendermaßen: Die Breite und Höhe übernehmen wir vom Ausgangsblock, aber wir setzen die Koordinaten auf X=Y=0. Sie müssen sich also vorstellen, wir haben uns im Speicher einen Bildspeicher genau in der Größe eines Blocks erstellt, in den wir nun unseren genau passenden Ausschnitt hineinkopieren, also an die Koordinate (0,0). Es wäre natürlich auch möglich, einen kleineren Ausschnitt an eine andere Koordinate zu kopieren, aber dies paßt nicht zu unserem Beispiel. Also besetzen wir das punkte[]-Feld folgendermaßen (siehe Abb. 4).

Bevor wir jetzt zum eigentlichen Kopieren kommen, muß noch errechnet werden, wie groß der Speicherbedarf des Blocks sein darf. Dies ist ganz einfach:

bytes = Anzahl der Worte pro Zeile * 2 * Höhe * Anzahl des Planes;

ziel.fd__addr = Malloc(bytes);

Jetzt können Sie kopieren! Aber was ist nun schon wieder der Modus, den wir bisher immer auf 3 gesetzt haben? Dieser gibt die Verknüpfung zwischen zu kopierendem Ausschnitt (Ausgang) und Hintergrund (Ziel) an. Der Verknüpfungsmodus kann mit vier Bits eingestellt werden, wobei dann jedes Bit einer Verknüpfungsart entspricht und bei mehreren gesetzten Bits diese Verknüpfungen dann miteinander UND-verknüpft werden:

A bedeutet Ausgangspixel
~A bedeutet Ausgangspixel (invertiert)
Z bedeutet Zielpixel (alter Bildpunkt)
~Z bedeutet Zielpixel (invertiert)
| ODER-VERKNÜPFUNG
& UND-Verknüpfung
^EXOR-Verknüpfung

Bit 0: Ergebnis = A & Z
Bit 1: Ergebnis = A & ~Z
Bit 2: Ergebnis = ~A & Z
Bit 3: Ergebnis = ~(A | Z)

```
punkte[0] = 10;                  /* X-Koordinate Ausgang */
punkte[1] = 40;                  /* Y-Koordinate Ausgang */
punkte[2] = punkte[6] = 133;     /* Breite */
punkte[3] = punkte[7] = 120;     /* Höhe */
punkte[4] =   0;                 /* X-Koordinate Ziel */
punkte[5] =   0;                 /* Y-Koordinate Ziel */
```

Abb. 4

Ein Modus von 12 würde die Bitkombination 1100 und damit folgende End-Verknüpfung bedeuten: (siehe Abb. 5)

Ein zweites Beispiel mit Modus 7: (Abb. 6)

Anmerkung:
Die Umwandlungen sind mit Hilfe der Boolschen Algebra vollzogen worden. Zusammengefaßt in einer Tabelle sieht es folgendermaßen aus: (siehe Abb. 7)

Sie sehen also, daß es vielfältige Möglichkeiten gibt, Ausgang und Ziel zu verknüpfen.

Bevor ich Ihnen noch einige Tips gebe, wie Sie die Kopierroutinen optimal einsetzen können, möchte ich noch einmal auf das Beispiel-Programm eingehen: (siehe Listing 1)

Es handelt sich um ein kleines Programm, das es ermöglicht, Teile aus dem Bildschirm auszuschneiden und mehrmals auf dem Bildschirm zu kopieren. Um besser erkennen zu können, welche Werte was bedeuten, werden die Koordinaten zunächst in eine GRECT-Structur gebracht. Dadurch sind alle vier Koordinaten eines Ausschnittes in einer Variable zusammengefaßt.

Durch die Routine grect__to__array() werden die Werte der GRECT-Struktur wieder in ein Feld kopiert. Auf das Benutzen einer Dialogbox wurde übrigens verzichtet, da für diesen Zweck auch eine ALERT-Box 'mißbraucht' werden kann, und man damit einiges an Speicherplatz spart. Übrigens ist dieses Programm ganz lustig, wenn man es als Accessory ausführt, da man dann im DESKTOP ein wenig kopieren kann...

Hier nun ein paar Hinweise zum optimalen Einsatz der VDI-Kopierroutine:

a) Versuchen Sie soweit es geht, Koordinaten zu wählen, die auf eine Wortgrenze (X = 16, 32, 48...) fallen. Dadurch muß das Bild nicht noch erst intern bitweise verschoben werden, was sehr zeitintensiv ist, vor allem, wenn Sie Texte mit dem VDI ausgeben!

b) Vermeiden Sie Modi, in denen die Zielfläche erst noch gelesen werden muß. Wenn Sie auf solche Verknüpfungen verzichten, können Sie Zeiten von bis zu 50 % einsparen.

c) Vermeiden Sie unterschiedliche Verschiebungen einer Wortgrenze in Ausgang und Ziel, beispielsweise eine

```
Ergebnis = Bit3-Verknüpfung ! Bit4-Verknüpfung
Ergebnis = ~(A | Z) | ~A & Z -
         = ~A & ~Z | ~A & Z
         = ~A & (~Z | Z)
         = ~A & 1
         = ~A
```

Abb.5

```
Ergebnis = A & Z | ~A & Z | A & ~Z
         = Z & (A | ~A) | A & ~Z
         = Z & 1 | A & ~Z
         = Z | A & ~Z
         = Z | A & (Z | ~Z)
         = Z | A & 1
         = Z | A
```

Abb. 6

<table>
<tr><th>Modus</th><th>Verknüpfung</th><th>Anmerkung</th></tr>
<tr><td>0</td><td>E = 0</td><td>Löschen des Zielblocks (s. Anmerkund d)</td></tr>
<tr><td>1</td><td>E = A & Z</td><td></td></tr>
<tr><td>2</td><td>E = A & ~Z</td><td></td></tr>
<tr><td>3</td><td>E = A</td><td>Ersetzen des Ziels durch Ausgang</td></tr>
<tr><td>4</td><td>E = ~A & Z</td><td></td></tr>
<tr><td>5</td><td>E = Z</td><td>dann kann man es auch sein lassen</td></tr>
<tr><td>6</td><td>E = A Z</td><td>Pixel setzen, wenn A u. Z verschieden</td></tr>
<tr><td>7</td><td>E = A | Z</td><td>transparent, Ziel scheint durch</td></tr>
<tr><td>8</td><td>E = ~(A | Z)</td><td>wie 7, nur invertiert</td></tr>
<tr><td>9</td><td>E = ~(A ^ Z)</td><td>wie 6, nur invertiert</td></tr>
<tr><td>10</td><td>E = ~Z</td><td>Invertieren des Zielblocks (s. Anmerkung d)</td></tr>
<tr><td>11</td><td>E = A | ~Z</td><td></td></tr>
<tr><td>12</td><td>E = ~A</td><td>Ersetzen, mit invertierendem Ausgang</td></tr>
<tr><td>13</td><td>E = ~A | Z</td><td>invertiert, transparent</td></tr>
<tr><td>14</td><td>E = ~(A & Z)</td><td></td></tr>
<tr><td>15</td><td>E = 1</td><td>Füllen (s. Anmerkung d)</td></tr>
</table>

Abb. 7

```
vrt__cpyfm(vdi__handle, modus, punkte, ausgang, ziel, farben);
Die Parameter sind bis auf das Feld farben[] schon oben erklärt:
int farben[2]; /* farbe[0] = Farbe der gesetzten Punkte */
               /* farbe[1] = Farbe der ungesetzten Punkte */
```

Abb. 8

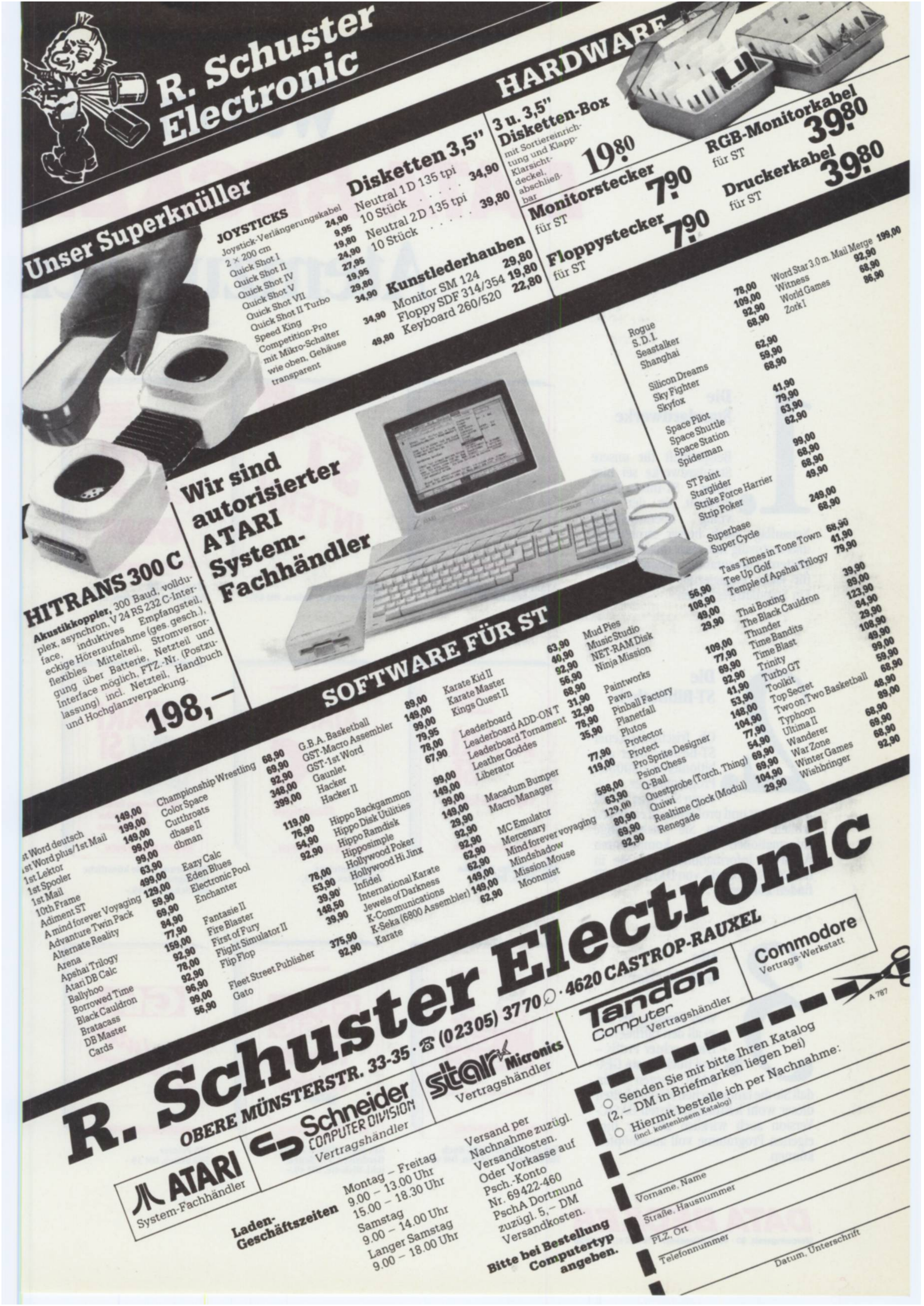
X-Koordinate von 4 im Ausgang und 9 im Ziel. Dadurch muß das Ausgangsbild erst um 5 Pixel verschoben werden.

d) Das Füllen, Löschen und Invertieren geht mit der Routine v__bar() schneller.

Die Routine **vrt__cpyfm()** unterscheidet sich um eine wesentliche Kleinigkeit von der Routine vro__cpyfm(). Mit vro__cpyfm() ist es nicht möglich, ein Monochrombild in eine Farbauflösung zu kopieren. Dies gelingt aber mit vrt__cpyfm(). Dabei können Sie noch zusätzlich angeben, welche Farbe ein gesetzter Punkt und ein ungesetzter Punkt haben soll. Folgende Parameter ergeben sich: (siehe Abb. 8).

Mit diesen Angaben können Bildschirmausschnitt kopiert und abgespeichert werden. Wenn Sie die Bilder auf der Diskette abspeichern, so denken Sie daran auch den FDB mit abzuspeichern.

Stefan Höhn


R. Schuster
Electronic
HARDWARE
Unser Superknüller
JOYSTICKS
Joystick-Verlängerungskabel
2 × 200 cm 24,90
Quick Shot I 9,95
Quick Shot II 19,80
Quick Shot IV 24,90
Quick Shot V 27,95
Quick Shot VII 19,95
Quick Shot II Turbo 29,80
Speed King 34,90
Competition-Pro
mit Mikro-Schalter 34,90
wie oben, Gehäuse
transparent 49,80
Disketten 3,5"
Neutral 1 D 135 tpi
10 Stück 34,90
Neutral 2 D 135 tpi
10 Stück 39,80
Kunstlederhauben
Monitor SM 124 29,80
Floppy SDF 314/354 19,80
Keyboard 260/520 22,80
3 u. 3,5"
Disketten-Box
mit Sortiereinrichtung und Klapp-Klarsichtdeckel, abschließbar
19.80
Monitorstecker
für ST
7.90
Floppystecker
für ST
7.90
RGB-Monitorkabel
für ST
39.80
Druckerkabel
für ST
39.80
Wir sind autorisierter ATARI System-Fachhändler
HITRANS 300 C
Akustikkoppler, 300 Baud, vollduplex, asynchron, V 24 RS 232 C-Interface, induktives Empfangsteil, eckige Höreraufnahme (ges. gesch.), flexibles Mittelteil, Stromversorgung über Batterie, Netzteil und Interface möglich, FTZ.-Nr. (Postzulassung) incl. Netzteil, Handbuch und Hochglanzverpackung.
198,—
SOFTWARE FÜR ST
1st Word deutsch 149,00
1st Word plus/1st Mail 199,00
1st Lektor 149,00
1st Spooler 99,00
1st Mail 99,00
10th Frame 63,90
Adiment ST 499,00
A mind forever Voyaging 129,00
Advanture Twin Pack 59,90
Alternate Reality 69,90
Arena 84,90
Apshai Trilogy 77,90
Atari DB Calc 159,00
Ballyhoo 92,90
Borrowed Time 78,00
Black Cauldron 92,90
Bratacass 96,90
DB Master 99,00
Cards 56,90
Championship Wrestling 68,90
Color Space 69,90
Cutthroats 92,90
dbase II 348,00
dbman 399,00
Eazy Calc 119,00
Eden Blues 76,90
Electronic Pool 54,90
Enchanter 92,90
Fantasie II 78,00
Fire Blaster 53,90
First of Fury 39,90
Flight Simulator II 148,50
Flip Flop 39,90
Fleet Street Publisher 375,90
Gato 92,90
G.B.A. Basketball 89,00
GST-Macro Assembler 149,00
GST-1st Word 99,00
Gaunlet 79,95
Hacker 78,00
Hacker II 67,90
Hippo Backgammon 99,00
Hippo Disk Utilities 149,00
Hippo Ramdisk 99,00
Hipposimple 149,00
Hollywood Poker 29,90
Hollywood Hi Jinx 92,90
Infidel 92,90
International Karate 62,90
Jewels of Darkness 62,90
K-Communications 149,00
K-Seka (6800 Assembler) 149,00
Karate 62,90
Karate Kid II
Karate Master
Kings Quest II
Leaderboard
Leaderboard ADD-ON T
Leaderboard Tornament
Leather Goddes
Liberator
63,90
40,90
92,90
56,90
68,90
31,90
32,90
78,90
35,90
Macadum Bumper 77,90
Macro Manager 119,00
MC Emulator 598,00
Mercenary 63,90
Mind forever voyaging 129,00
Mindshadow 80,90
Mission Mouse 69,90
Moonmist 92,90
Mud Pies 56,90
Music Studio 108,90
NET-RAM Disk 49,00
Ninja Mission 29,90
Paintworks 109,00
Pawn 77,90
Pinball Factory 69,90
Planetfall 92,90
Plutos 41,90
Protector 53,90
Protect 148,00
Pro Sprite Designer 104,90
Psion Chess 77,90
Q-Ball 54,90
Questprobe (Torch, Thing) 69,90
Quiwi 69,90
Realtime Clock (Modul) 104,90
Renegade 29,90
Rogue 78,00
S.D.I. 109,00
Seastalker 92,90
Shanghai 68,90
Silicon Dreams 62,90
Sky Fighter 59,90
Skyfox 68,90
Space Pilot 41,90
Space Shuttle 79,90
Space Station 63,90
Spiderman 62,90
ST Paint 99,00
Starglider 68,90
Strike Force Harrier 68,90
Strip Poker 49,90
Superbase 249,00
Super Cycle 68,90
Tass Times in Tone Town 68,90
Tee Up Golf 41,90
Temple of Apshai Trilogy 79,90
Thai Boxing 39,90
The Black Cauldron 89,00
Thunder 123,90
Time Bandits 84,90
Time Blast 29,90
Trinity 108,90
Turbo GT 49,90
Toolkit 99,00
Top Secret 59,90
Two on Two Basketball 68,90
Typhoon 48,90
Ultima II 89,00
Wanderer 68,90
War Zone 69,90
Winter Games 68,90
Wishbringer 92,90
Word Star 3.0 m. Mail Merge 199,00
Witness 92,90
World Games 68,90
Zork I 86,90
R. Schuster Electronic
OBERE MÜNSTERSTR. 33-35 · ☎ (02305) 3770 · 4620 CASTROP-RAUXEL
ATARI
System-Fachhändler
Schneider
COMPUTER DIVISION
Vertragshändler
star Micronics
Vertragshändler
Tandon Computer
Vertragshändler
Commodore
Vertrags-Werkstatt
Laden-Geschäftszeiten
Montag – Freitag
9.00 – 13.00 Uhr
15.00 – 18.30 Uhr
Samstag
9.00 – 14.00 Uhr
Langer Samstag
9.00 – 18.00 Uhr
Versand per Nachnahme zuzügl. Versandkosten.
Oder Vorkasse auf Psch.-Konto Nr. 69422-460 PschA Dortmund zuzügl. 5,– DM Versandkosten.
Bitte bei Bestellung Computertyp angeben.
○ Senden Sie mir bitte Ihren Katalog (2,– DM in Briefmarken liegen bei)
○ Hiermit bestelle ich per Nachnahme:
(incl. kostenlosem Katalog)
Vorname, Name
Straße, Hausnummer
PLZ, Ort
Telefonnummer
Datum, Unterschrift
A 787

Warum AT
DATA BECKER
Atemzug gen

1. Die Standardwerke

Beispielhaft für unsere Standardwerke sei hier unser ST-Intern-Band genannt. In der jetzt völlig überarbeiteten Neuauflage noch besser strukturiert und erstmalig mit einer ausführlichen Blitterdokumentation. Unentbehrlich für jeden engagierten ST-Anwender. Ein Standardwerk eben.



ATARI ST für Einsteiger
248 Seiten, DM 29,–



ATARI ST Intern
Hardcover, 637 Seiten, DM 69,–



ATARI ST GEM
Hardcover, 691 Seiten, DM 69,–

2. Die ST-Bibliothek

Ob frischgebackener ST-Besitzer oder ambitionierter 68000er-Programmierer – wenn Sie Ihren ATARI ST effizient und professionell einsetzen wollen, brauchen Sie hochkarätige Informationen von kompetenten Autoren. Informationen, die Sie in der „ST-Bibliothek" von DATA BECKER finden können.



ATARI ST Tips & Tricks
352 Seiten, DM 49,–



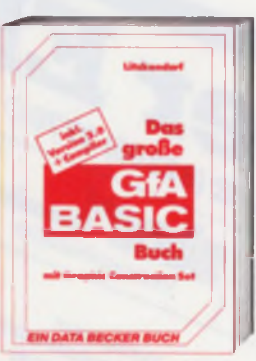
C für Einsteiger
393 Seiten, DM 39,–



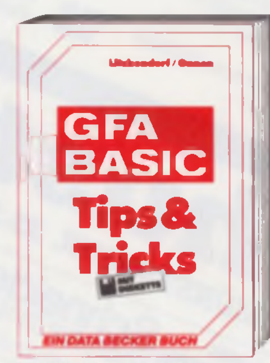
Einführung in die künstliche Intelligenz
406 Seiten, dM 49,–

3. Die GFA-Bücher

Vom Einstieg bis hin zu all den raffinierten Tricks echter Profis – drei Bücher zum GFA-BASIC sorgen dafür, daß Sie die fantastischen Möglichkeiten dieser wohl leistungsstärksten BASIC-Version auch wirklich alle für Ihre eigenen Programme voll ausschöpfen können.



Das große GFA-BASIC-Buch
Hardcover, 574 Seiten, DM 49,–



GFA-BASIC Tips & Tricks
Hardcover, 350 Seiten, inkl. Diskette, DM 49,–



GFA-Painter
382 Seiten, DM 39,–

ARI ST und
meist im gleichen
annt werden.



ARI ST Floppy und Harddisk
ardcover, 522 Seiten, DM 59,–

4. Alles zum Thema Grafik

Setzen Sie die Grafikfähigkeiten Ihres ST gezielt für Ihre eigenen Anwendungen ein. Diese Bücher zeigen Ihnen, was alles möglich ist. Von einer flackerfreien Animation bis hin zu atemberaubenden 3-D-Grafiken finden Sie hier das gesamte Know-how zum Thema Grafik.



Das Supergrafikbuch zum ATARI ST
Hardcover, 838 Seiten, inkl. Diskette, DM 69,–



3-D-Programmierung
Hardcover, 601 Seiten, inkl. Diskette, DM 69,–



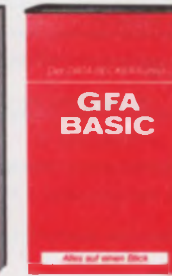
as Maschinensprachebuch zum
ARI ST
ardcover, 334 Seiten, DM 39,–

5. Die DATA BECKER Führer

Kompakte Informationsquellen, die den Anwender bei seiner Arbeit mit dem ST nicht allein lassen. Auf einen Blick findet er alle wichtigen Kommandos und Befehle. Schnell und zuverlässig. Für die tägliche Arbeit am Rechner einfach unentbehrlich.



Der DATA BECKER Führer zum ATARI ST
240 Seiten, DM 29,80



Der DATA BECKER Führer zu GFA-BASIC
254 Seiten, DM 24,80



Der DATA BECKER Führer zu 1st Word
176 Seiten, DM 24,80

6. Programme für jeden Zweck

Jahrelange Erfahrung der DATA BECKER Programmierer macht es möglich: Spitzenprogramme, mit denen jeder arbeiten kann. Komfortabel, leistungsfähig und vielseitig. Zu einem Preis, der sich sehen lassen kann. Hier ein kleiner Ausschnitt:

TEXTOMAT ST – die ideale Textverarbeitung für Ihren ST. Mit allen Leistungsmerkmalen einer modernen Textverarbeitung und einem Höchstmaß an Komfort. **DM 99,–**

DATAMAT ST – das maßgeschneiderte Dateiprogramm. Glänzt mit hoher Geschwindigkeit beim Suchen, Blättern und Editieren. **DM 99,–**

BECKERtext ST – mehr als eine gewöhnliche Textverarbeitung. Mit Rechnen im Text, einem individuell erweiterbaren ONLINE-Lexikon, einer professionellen Formularverwaltung und mehrspaltiger Druckausgabe. **DM 199,–**

BECKERbase – ein komplettes, leistungsstarkes ST-Datenbankprogramm für unter 100 Mark. Individuell lassen sich umfassende Datenbank-Anwendungen an eigene Wünsche anpassen. **DM 99,–**

COUPON

An: DATA BECKER · Merowingerstr. 30
4000 Düsseldorf
Bitte senden Sie mir:

zzgl. DM 5,– Versandkosten
unabhängig von der bestellten Stückzahl
☐ per Nachnahme ☐ Verrechnungsscheck liegt bei

Name

Straße

Ort

```c
#include <obdefs.h>

#define ARROW       0     /* Gemdefinierungen für die Maus */
#define M_OFF     256
#define M_ON      257

#define BUFLEN 33000      /* Länge des Buffers zum Abspeichern */
#define ERSETZEN 3        /* Modi für das Schreiben in den Bild */
#define EXODIEREN 6
#define ODIEREN 7

#define END 3             /* Die Knöpfe der Auswahlbox */
#define PASTE 2
#define CUT 1

typedef struct grect      /* Bildkoordinaten-Struktur */
{
        int     g_x;
        int     g_y;
        int     g_w;
        int     g_h;
} GRECT;

typedef struct fdbstr     /* Form-Definition-Block-Struktur */
{
        long            fd_addr;      /* Adresse der Bilddaten */
        int             fd_w;         /* Breite des Bildes */
        int             fd_h;         /* Höhe des Bildes */
        int             fd_wdwidth;   /* Breite in Worten */
        int             fd_stand;     /* reserviert immer 1*/
        int             fd_nplanes;   /* Anzahl der Farbebenen->Planes */
        int             fd_r1;        /* reserviert */
        int             fd_r2;        /* reserviert */
        int             fd_r3;        /* reserviert */
} FDB;

GRECT ausgang1      = { 0,0,0,0 };   /* Koordinaten des Ausschnitts */
GRECT ziel          = { 0,0,0,0 };   /* Koordinaten der Kopie */
FDB bildschirm      = { 0x0L,640,400,40,0,1,0,0,0 }; /* Ausgangs-FDB */
FDB block1          = { 0x0l,400,400,25,0,1,0,0,0 };  /* Ziel-FDB */

GRECT *ausg;                  /* Zeiger auf Ausgangsblockstruktur */
char *buffer1;                /* Zeiger auf Bildbereich */
FDB  *blocker;                /* Zeiger auf FDB */

                      /* Vdi-Variablen */
int vdi_handle;
int contrl[12], intin[128], intout[128], ptsin[128], ptsout[128];
extern int gl_apid;
int work_in[11];
int work_out[57];

int planes;                   /* Anzahl des Planes */
int mode;                     /* Kopiermodus */

main()
{
   int exit_obj= PASTE;

   appl_init();               /* Applikation initialisieren */
   open_vwork();              /* VDI-Workstation inititialisieren */

   graf_mouse(ARROW,0l);    /* Pfeil als Maus einschalten */

   block1.fd_addr=(long)buffer1; /* Bild-Speicheradresse */

   ausg=&ausgang1;            /* Zeiger auf Variable zeigen lassen */
   blocker=&block1;           /* dito */

   ausgang.g_w=ausg->g_w; /* Breite in Struktur setzen */
   ausgang.g_h=ausg->g_h; /* Höhe setzen */
   ausgang.g_x=x;             /* Ausgangskoordinaten setzen */
   ausgang.g_y=y;
   ausg->g_x=0;               /* Im Ausgangsfeld (fest)immer 0 */
   ausg->g_y=0;               /* dito */

   grect_to_array(ausg,punkte);     /* Umwandlung von GRECT->Feld */
   grect_to_array(&ausgang,&punkte[4]);/* dito in zweiten Teil vom Feld */

      /* Raster-Kopieren Speicher -> Bildschirm */

   vro_cpyfm(vdi_handle,ERSETZEN,punkte,blocker,&bildschirm);

   graf_mouse(M_ON,0l);    /* Maus, laß dich sehen ! */
}

do_cut() /* Ermitteln der Koordinaten des Ausschnitts und Ausschneiden */
{
   int x, y, breite, hoehe, dummy;

     /* Warten bis die Maustaste gedrückt ist */
   evnt_button(1,1,1,&x,&y,&dummy,&dummy);

     /* GUMMIBAND-BOX mit x,y-Kordinaten zeichnen */
     /* und die gewünschte Auschnittsgröße ermitteln */
   graf_rubberbox(x,y,1,1,&breite,&hoehe);

   cut(x,y,breite,hoehe,1); /* Auschnitt ausschneiden */
}

cut(x,y,w,h, fl) /* Ausschnitt in anderen Speicherbereich kopieren */
unsigned int x,y,w,h; /* Koordinaten */
{
   register int yh;
   int punkte[8];

   graf_mouse(M_OFF,0l);    /* Maus, aus! */

     ausg=&ausgang1;      /* Zeiger auf Variable zeigen lassen */
     blocker=&block1;

/* Pufferspeicher löschen */
  for(yh=0; yh<BUFLEN; buffer1[yh++]=0);

  ausg->g_x=x;            /* Koordinaten setzen */
  ausg->g_y=y;
  ziel.g_w=ausg->g_w=w;
  ziel.g_h=ausg->g_h=h;
  ziel.g_x=0; /* ist immer 0 */
  ziel.g_y=0;

  blocker->fd_wdwidth= ((w/16)+1); /* Breite in Worten errechnen */
  blocker->fd_w = w;
  blocker->fd_h = h;
  blocker->fd_addr = (long)buffer1;/* Speicheradresse setzen */

  grect_to_array(ausg,punkte);      /* Umwandlung von GRECT->Feld */
  grect_to_array(&ziel,&punkte[4]);/* dito in zweiten Teil vom Feld */

  /* Raster-Kopieren Bildschirm->Speicher */
  vro_cpyfm(vdi_handle,ERSETZEN,punkte,&bildschirm,blocker);

 ausg=&ausgang1;
 blocker=&block1;

 buffer1=(char*)(Malloc((long)BUFLEN));  /* Speicher reservieren */
 if (buffer1==(char*)0)                  /* Nicht genug vorhanden */
 {
    form_alert(1,"[1][Nicht genug Speicher vorhanden!][ OK ]");
    v_clsvwk();                          /* Workstation schließen */
    appl_exit();                         /* Applikation beenden */
    gemdos(0);                           /* Programm beenden */
 }
 while(exit_obj!=END)     /* Bis die Taste END gedrückt ist */
 {

     wind_update(1);      /* eigene Bildschirmausgabe- kein REDRAW */
     wind_update(3);      /* eigene Maussteuerung ohne Menübegrenzung */

     planes=4>>Getrez(); /* Auflösung und Planes bestimmen */
     block1.fd_nplanes=bildschirm.fd_nplanes=planes; /* einschreiben */

     /* Auswahl der Funktion, die Box wurde als ALERT-Box gewählt */

     exit_obj=form_alert(0,"[1][Wählen Sie bitte aus:][Schere|Kleber|Ende]");

     switch(exit_obj)     /* Welche Taste war's wohl  ? */
     {
        case PASTE:             /* Einfugen */
           do_paste();          /* los geht's */
           break;

        case CUT:
           do_cut();            /* Ausschneiden */
           break;               /* ab geht's */

     }
     wind_update(2);   /* Mäuschen darf wieder von GEM geführt werden */
     wind_update(0);   /* Auch die Bildschirmausgabe geben wir zurück */
 }
 Mfree(buffer1);           /* Speicherbereich freigeben */
 v_clsvwk(vdi_handle); /* VDI-Workstation schließen */
 appl_exit();              /* Applikation beenden */
}                          /* Auf Wiedersehen, bis zum nächsten Mal */

do_paste() /* Routine, um Einfüge-Koordinate herauszufinden*/
{          /* und dann zu Kopieren */
   int retx,rety,mx,my,w,h;

   mode=ERSETZEN;                  /* kann nach Belieben geändert werden */

       /* Warten bis die Maustaste gedruckt worden ist */
   evnt_button(1,1,1,&mx,&my,&retx,&retx);

       /* Box in der richtigen Größe verschieben */
   graf_dragbox(ausg->g_w,ausg->g_h,mx,my,0,0,640,400,&retx,&rety);

   paste(retx,rety,w,h,mode);    /* Laßt es uns kopieren */
}

paste(x,y,w,h,modus)  /* Kopierroutine */
unsigned int x,y,w,h,modus; /* Kordinaten und Einfügemodus */
{
   GRECT ausgang; /* Koordinaten-Struktur */
   int punkte[8];

   graf_mouse(M_OFF,0l);  /* Maus beim Kopieren ausschalten */


   graf_mouse(M_ON,0l);  /* Maus darf wieder erscheinen */
}

/* Umwandlungsroutine: von GRECT in FELD schreiben */
grect_to_array(rechteck,array)
GRECT *rechteck;
int *array;
{
   *array++ = rechteck->g_x;
   *array++ = rechteck->g_y;
   *array++ = rechteck->g_x + rechteck->g_w - 1;
   *array++ = rechteck->g_y + rechteck->g_h - 1;
}

open_vwork() /* Initialisieren der VDI-Workstation */
{
   int i;

   for (i=0; i<10; work_in[i++]);
   work_in[10]=2;
   v_opnvwk(work_in, &vdi_handle, work_out);
}
```

Der Fehlerteufel geht um

Leider haben sich im Listing 'Interrupts in Megamax C' einige Fehler in Abb. 6 eingeschlichen. Die neunte Zeile muß folgendermaßen lauten:

static void__timavec()

Der Xbtimer-Aufruf in der vorletzten Zeile lautet:

Xbtimer (TIMER__A,TIMACTRL, TIMADAT,__timavec);

Abb. 6 ersetzt im Hauptlisting Zeile 23...44. Falls Sie die kleinen Unstimmigkeiten nicht schon selbst entdeckt haben, so hoffen wir, daß es nun funktioniert.



Programmier Praxis

In der neuen Rubrik 'PROGRAMMIER PRAXIS' wollen wir ausschließlich kurze Programme zur praktischen Anwendung vorstellen. Wir kommen damit der großen Nachfrage der Leser nach.

Um einen interessanten Inhalt und eine ausgeglichene Berücksichtigung aller Sprachen zu gewährleisten möchten wir Sie bitten, sich an dieser Rubrik mit eigenen Programmen und Routinen zu beteiligen. Veröffentlichte Artikel werden natürlich angemessen honoriert.

INHALT

Der Wievielte ist heute?

DATUM per Mausklick

Sicher haben Sie sich schon gefragt, ob man diese kleinen Uhrplatinen, die man zusätzlich in den ST einbauen kann, nicht intensiver nutzen kann. Nur um immer das richtige Datum auf den Disketten zu haben, dazu sind sie zu teuer. Man müßte zum Beispiel das Datum in einer Textverarbeitung o. ä. an der Stelle des Cursors ausgeben lassen können. Wenn Sie diese Funktion in herkömmlichen Textverarbeitungsprogrammen vermissen, dann ist dieses mit dem Megamax C-Compiler geschriebene Accessory genau das richtige für Sie.

Natürlich funktioniert das Programm auch dann, wenn man keine zusätzliche Uhr eingebaut hat. Man muß nur einmal das Datum mit Hilfe des Kontrollfeldes eingeben. Das Datum kann auf zwei verschiedene Arten ausgegeben werden: kurz („5.6.1987“) und lang („5. Juni 1987“). Nachdem man den entsprechenden Eintrag im Menü angeklickt hat, erscheint eine Alertbox (siehe Bild 1). Das Datum wird anschließend hinter dem Cursor ausgegeben (Bild 2).

Das Besondere an dem Programm ist, daß es den Tastaturpuffer verändert. Dazu wird die Routine „output()“ benutzt. Dieses Unterprogramm speichert einen vorher zusammengestellten String in den Tastaturpuffer. Bevor man diese Routine in eigene Programme einbauen kann (was natürlich möglich ist), muß man einiges wissen:

Das Betriebssystem des ST besitzt eine Funktion namens „Iorec“ (Xbios 14), welche einem die Adresse des Tastaturpuffers und andere interessante Parameter übergibt (besser: sie übergibt die Adresse auf eine Tabelle mit den gewünschten Werten). Der Tastaturpuffer ist 256 Byte lang (auch das erfahren wir durch diese Funktion) und kann 64 Zeichen aufnehmen. Das bedeutet, daß pro Zeichen vier Byte gebraucht werden. Im ersten Byte sind einige Flags gespeichert, die angeben, ob z. B Control oder Alternate zusammen mit der Taste gedrückt wurden. Das zweite Byte gibt den Scancode der Taste an. Das dritte Byte ist ganz einfach leer und im vierten Byte schließlich steht der Ascii-Code der gedrückten Taste (Bild 3). Es existieren noch zwei Zeiger, die auf den Anfang und das Ende der noch auszugebenden Zeichen zeigen. Das bedeutet, daß wir es mit einem „Fifo“-Puffer (first in-first out) zu tun haben. Wenn diese Zeiger den gleichen Wert haben, ist der Puffer leer. Wenn nicht, werden die Zeichen nacheinander ausgegeben.

Jetzt wird's kompliziert: Die „output()“-Routine schreibt nicht sofort alle Zeichen in den Tastaturpuffer. Es gibt Programme (1st Word Plus), die den Tastaturpuffer gerne mal leeren, und dann wäre vom Datum nicht mehr viel übrig. Also muß ein Buchstabe nach dem anderen in den Tastaturpuffer geschrieben und dann von der Textverarbeitung ausgegeben werden. Anders gesagt: Der ST muß zwei Programme quasi gleichzeitig abarbeiten (Multitasking), wobei ein Programm (die Textverarbeitung) gar nicht dafür vorbereitet ist. Die Lösung des Problems heißt: EVNT__TIMER! Nachdem dieser AES-Befehl abgearbeitet wurde, wird die Textverarbeitung gestartet. Diese wiederum wartet auf ein Zeichen im Tastaturpuffer, indem sie ebenfalls einen EVNT-Befehl abarbeitet. Nun tritt das Accessory wieder in Aktion. Ein Zeichen wird in den Tastaturpuffer geschrieben und der EVNT__TIMER-Befehl erneut abgearbeitet. Die Textverarbeitung findet das Zeichen im Puffer und gibt es aus. Die Suche nach einem weiteren Zeichen ist erfolglos, und so wird das Accessory erneut gestartet. Nachdem alle Zeichen ausgegeben wurden, wird diese Routine beendet.

Ein Tip für 1st Word Plus-Benutzer: Wenn Sie das Datum sauber ausgeben wollen, dann schreiben Sie zuerst den Ort, geben das Datum aus und wählen dann im Menüpunkt „Attribute“ die Funktion „rechtsbündig“ aus.

Simon Gleissner



Bild 1: Das ist die Eingabe

Dies ist ein Test: 5. Juni 1987
Dies ist auch ein Test: 5.6.1987

Bild 2: Und so sieht es aus



```
  1:  /*******************************************************/
  2:  /*                                                     */
  3:  /*                    D A T U M  V1.0                  */
  4:  /*                                                     */
  5:  /* geschrieben von: Simon Gleissner                    */
  6:  /*                  Bingenheimer Str. 46               */
  7:  /*                  6361 Reichelsheim                  */
  8:  /*                                                     */
  9:  /*                    mit MEGAMAX-C                    */
 10:  /*******************************************************/
 11:
 12:  #include "osbind.h"  /* alles laden... */
 13:  #include "obdefs.h"
 14:  #include "gemdefs.h"
 15:  #include "define.h"
 16:  #include "gembind.h"
 17:
 18:  int  contrl[12],      /* diverse Variablen & Felder */
 19:       intin[128],
 20:       ptsin[128],
 21:       intout[128],
 22:       ptsout[128],
 23:       msgbuff[8],
 24:       menu_id,
 25:       alert,
 26:       datum,
 27:       schleife,
 28:       monat,
 29:       out_pointer,
 30:       ibufsize,
 31:       ibufhd,
 32:       ibuftl;
 33:  long iorec_point,
 34:       ibuf;
 35:  char out_puffer[30];
 36:  char fo_astring[]=
 37:   "[1][DATUM V1.0 wurde von|Simon Gleissner geschrieben.|Wie möchten Sie das Datu
       m?|Lang: 24. Dezember 1987|Kurz: 24.12.1987][Lang|Kurz|Zurück]";
 38:  /* Bitte beim Abtippen beachten: Der String muß in einer Zeile stehen! */
 39:  extern gl_apid;
 40:
 41:  main()                              /* Hauptprogramm */
 42:  {
 43:    iorec_point=Iorec(1);             /* Zeiger auf Tastaturzeiger */
 44:    ibuf=*(long*)iorec_point;         /* Zeiger auf Tastaturpuffer */
 45:    ibufsize=*(int*)(iorec_point+4);  /* Größe des Puffers */
 46:    appl_init();                      /* Applikation anmelden */
 47:    menu_id=menu_register(gl_apid,"  Datum "); /* Accessory anmelden */
 48:    FOREVER                           /* Endlosschleife */
 49:    {
 50:      out_pointer=0;          /* Zeiger an den Anfang des  Ausgabepuffer */
 51:      evnt_mesag(msgbuff);    /* Auf Nachricht im Ereignispuffer warten */
 52:      if(msgbuff[0]==AC_OPEN&&msgbuff[4]==menu_id) /* Accessory angeklickt? */
 53:      {
 54:        alert=form_alert(3,fo_astring); /* Alertbox anzeigen */
 55:        if(alert!=3)                    /* kein Abbruch */
 56:        {
 57:          date();             /* Datum in Ausgabepuffer schreiben */
 58:          output();           /* Ausgabepuffer ausgeben */
 59:        }
 60:      }
 61:    }
 62:  }
 63:
 64:  output()   /* Ausgabepuffer ausgeben */
 65:  {          /* (jetzt kommt ein bißchen Multitasking) */
 66:    for(schleife=0;schleife<out_pointer;) /* Schleife */
 67:    {
 68:      evnt_timer(0,0);        /* Kontrolle des Computers kurzzeitig abgeben */
 69:      ibufhd=*(int*)(iorec_point+6);   /* Head-Zeiger des Tastaturpuffers */
 70:      ibuftl=4+*(int*)(iorec_point+8); /* Tail-Zeiger des Tastaturpuffers */
 71:      if(ibuftl==ibufsize)             /* zeigt Tail-Zeiger auf Pufferende? */
 72:        ibuftl=0;                      /* Ja: auf Pufferanfang */
 73:      if(ibuftl!=ibufhd)               /* Head=Tail? (Puffer voll) */
 74:      {                                /* Nein: Zeichen in Puffer schreiben */
 75:        *(long*)(ibuf+ibuftl)=0x1e0000+out_puffer[schleife++];
 76:        *(int*)(iorec_point+8)=ibuftl; /* und Tail abspeichern */
 77:      }
 78:    }
 79:  }
 80:
```

```
 81:  date() /* Datum in Ausgabepuffer schreiben */
 82:  {
 83:    datum=Tgetdate();         /* Datum holen, */
 84:    transform(datum & 31);    /* Tag isolieren & in den Puffer schreiben, */
 85:    to_buffer(46);            /* '.' in den Puffer schreiben und */
 86:    monat=(datum&480)/32;     /* Monat holen */
 87:    if(alert==1)              /* Wurde 'langes' Datum gewünscht */
 88:    {                         /* Ja: */
 89:      switch(monat)           /* Entsprechenden Monatsnamen in den Puffer */
 90:      {
 91:        case 1:   text(" Januar ");
 92:                  break;
 93:        case 2:   text(" Februar ");
 94:                  break;
 95:        case 3:   text(" März ");
 96:                  break;
 97:        case 4:   text(" April ");
 98:                  break;
 99:        case 5:   text(" Mai ");
100:                  break;
101:        case 6:   text(" Juni ");
102:                  break;
103:        case 7:   text(" Juli ");
104:                  break;
105:        case 8:   text(" August ");
106:                  break;
107:        case 9:   text(" September ");
108:                  break;
109:        case 10:  text(" Oktober ");
110:                  break;
111:        case 11:  text(" November ");
112:                  break;
113:        case 12:  text(" Dezember ");
114:                  break;
115:      }
116:    }
117:    else                    /* 'kurzes' Datum */
118:    {
119:      transform(monat);     /* Monat und */
120:      to_buffer(46);        /* '.' in den Puffer schreiben */
121:    }
122:    transform((datum&65024)/512+1980); /* Jahr isolieren und in den Puffer */
123:  }
124:
125:  text(string)              /* schreibt einen String in den Ausgabepuffer */
126:  char *string;
127:  {
128:    for(schleife=0;string[schleife]!=0;)
129:      to_buffer(string[schleife++]);
130:  }
131:
132:  transform(number)         /* wandelt eine Zahl in einen String und */
133:  int number;               /* schreibt diesen in den Ausgabepuffer */
134:  {
135:    int step=1000,
136:        flag=0,
137:        zahl;
138:
139:    for(schleife=0;schleife<4;schleife++)
140:    {
141:
142:      zahl=number/step;
143:      number-=zahl*step;
144:      step/=10;
145:      if(flag==1||zahl!=0)
146:      {
147:        to_buffer(48+zahl);
148:        flag=1;
149:      }
150:    }
151:  }
152:
153:  to_buffer(ascii) /* Schreibt ein einzelnes Zeichen in den Ausgabepuffer */
154:  char ascii;
155:  {
156:    out_puffer[out_pointer++]=ascii;
157:  }
158:
```

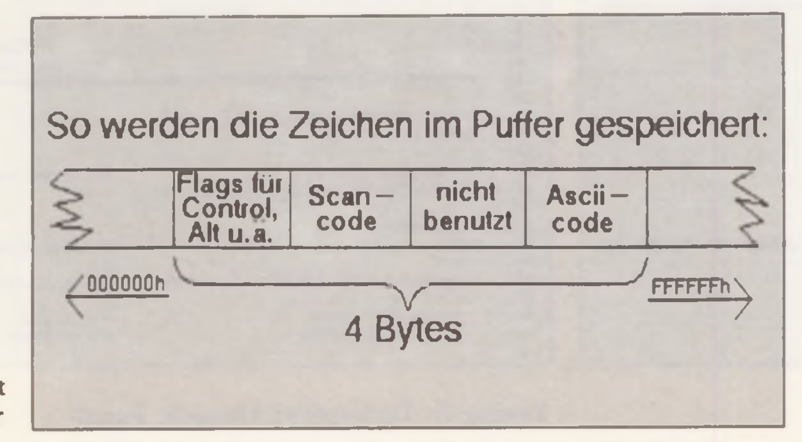


Bild 3: Das macht der Puffer



Genau genommen

Taylorentwicklung für Sinus und Cosinusfunktionen

Um numerische Aufgaben zu lösen, ist es oft nötig, bis auf die letzte Stelle zu rechnen, um Folgefehler so gering wie möglich zu halten. Dies nützt allerdings nichts, wenn die implementierten Funktionen schon Fehler liefern, die 3 oder 4 10er Potenzen unterhalb der Rechnergenauigkeit liegen, wie es bei den Sinus- und Cosinusfunktionen von Pascal+ der Fall ist.

Einen Anhaltspunkt für die Ungenauigkeit hat man nur an den Stellen 0, pi/2, pi... (z. B. sin(pi/2)= 1.0000000032). Um im gesamten Intervall die Funktionen zu testen, ohne einen genauen Wert zu haben, kann man den Einheitskreis verwenden. Hier bilden der Sinus und der Cosinus zusammen mit dem Radius ein rechtwinkliges Dreieck. Nach dem Satz von Pythagoras folgt:

r=sqrt(sin^2+cos^2)

Da im Einheitskreis der Radius die Länge 1 hat, erhält man mit 1−r einen Fehler, der indirekt etwas über die Genauigkeit der Sinus- und der Cosinusfunktion aussagt. Dies ist im Programm SINTEST.PAS dargestellt. (Natürlich sollte die Wurzelfunktion auch auf ihre Genauigkeit überprüft werden). Nun zur Behebung dieses Mangels.

Aufbereiten des Arguments x: Bei einem Aufruf der cos(x) Funktion wird zu x pi/2 hinzugezählt und sin(x) aufgerufen, da die Cosinusfunktion nur eine um pi/2 verschobene Sinusfunktion ist. Bei einem Aufruf der sin(x) Funktion wird der x-Wert, der beliebig reell sein kann, auf einen Bereich von 0 bis pi/2 reduziert, und das Vorzeichen aufbewahrt. Dieses Intervall genügt vollkommen, um sämtliche Funktionswerte zu ermitteln.

Berechnung von sin(x) im Intervall [0..pi/2]: Die Funktion sns berechnet die Funktionswerte von Sinus durch Taylorreihenentwicklung für Sinus und Cosinus.

Diese lautet für sin x:
x − x^3/3! + x^5/5! − x^7/7! + − ...

und für cos x:
1 − x^2/2! + x^4/4! − x^6/6! + − ...

Da die einzelnen Glieder der Entwicklungen sehr schnell gegen Null streben, kann man den jeweils letzten berechneten Wert mit der Rechnergenauigkeit vergleichen, und hat somit ein relativ gutes Abbruchkriterium. Es wurde neben der Genauigkeit auch Wert auf eine erträgliche Geschwindigkeit gelegt, deshalb mag die Art der Programmierung etwas seltsam erscheinen. Da Taylorreihen am Intervallrand sehr viele Glieder (und auch Rechenzeit) benötigen, um die Funktion anzunähern, wurde im Bereich 0 <= x < pi/4 nach der sin Reihe entwickelt und im Bereich pi/4 <= x < pi/2 nach der cos-Reihe im Intervall [0..pi/4]. Bei einer Rechnergenauigkeit von 1e−12 heißt das maximal 8 Glieder. Um weitere Rechenzeit einzusparen, wurden die Koeffizienten der Reihen (1/i!) ausgerechnet und in Felder abgespeichert. Um Fehler zu vermeiden, werden die einzelnen Glieder zuerst in einem weiteren Feld abgespeichert und erst am Ende von der kleinsten Zahl aufwärts zusammengezählt. Wer sich nicht vorstellen kann, welche Fehler auftreten können, teste einmal die beiden kleinen Programme: Test 1 und Test 2

```
 1:  program sintest0;
 2:  {*****************************************************************}
 3:  (* Programm zum testen der Genauigkeit der Sinusfunktion.         *)
 4:  {*****************************************************************}
 5:  var df,pi,b,x1,x2:real;
 6:
 7:  (*$I sincos.pas*)
 8:
 9:  begin
10:    pi:=arctan(1)*4;
11:    b:=0;
12:    while b <= 2*pi do
13:    begin
14:      x1:=cos(b);
15:      x2:=sin(b);
16:      df:=x2-x1;
17:      writeln(b,'   r: ',sqrt(x1*x1+x2*x2),' Fehler:',1-sqrt(x1*x1+x2*x2));
18:      b:=b+1e-2;
19:    end;
20:  writeln(chr(7));
21:  end.
```

Listing 1: Testen der Genauigkeit

```
 1:  {*****************************************************************}
 2:  (*                                                                *)
 3:  (*        SIN und COS Routinen fuer Pascal unter Verwendung von   *)
 4:  (*                          Taylor Reihen.                        *)
 5:  (*                                                                *)
 6:  (*                                                  07.07.1987    *)
 7:  (*                                                  Peter Sadoni  *)
 8:  {*****************************************************************}
 9:
10:  function sns(x:real):real;
11:  forward;
12:
13:  function sin(x:real):real;                       (* Einsprung fuer sin x   *)
14:  (* Reduzierung von x auf den Bereich 0..pi/2                               *)
15:
16:  var vz:integer;
17:  begin                                            (* x liegt zwischen:      *)
18:    vz:=1;                                         (*                        *)
19:    x:=x-trunc(x/6.283185307180)*6.283185307180;   (* -2*pi und 2*pi         *)
20:    if x<0 then                                    (*                        *)
21:      x:=x+6.283185307180;                         (* 0 und 2*pi             *)
22:    if x>3.141592653590 then                       (* 0 und pi mit           *)
```

Listing 2: Taylorentwicklung in Pascal



```
23:     begin                                      (* vz = Vorzeichen von sin*)
24:       vz:=-1;                                  (*                        *)
25:       x:=x-3.141592653590;                     (*                        *)
26:     end;                                       (*                        *)
27:     if x>1.570796326795 then                   (*                        *)
28:       x:=3.141592653590 -x;                    (* 0 und pi/2             *)
29:     sin:=vz*sns(x);                            (* Aufruf von eigentl.    *)
30:   end;                                         (*  Sinusfunktion         *)
31:
32:
33:   function cos(x:real):real;                   (* Einsprung fuer cos x   *)
34:   (* Umwandlung von Cosinus in Sinus                                     *)
35:   begin
36:
37:     x:=x-1.570796326795;                       (* um pi/2 verschieben    *)
38:     cos:=-sin(x);                              (* Aufruf von sin x       *)
39:   end;
40:
41:
42:   function sns;
43:   (* eigentliche Sinusfunktion zur Berechnung von sin x mit 0 <= x <= pi/2   *)
44:   const eps=1e-12;                             (* Rechnergenauigkeit     *)
45:   type vektor = array[1..10] of real;
46:   var x2,hlp1,hlp2:real;
47:       n,i:integer;
48:       koeff,hlp3:vektor;
49:
50:     procedure sinbelegen(var sinkoeff:vektor);
51:     (* Belegen von sinkoeff mit den Koeffizienten der Taylor-Reihe fuer Sinus  *)
52:     begin
53:       sinkoeff[1]:=1;
54:       sinkoeff[2]:=-0.166666666667;            (* -1/6                   *)
55:       sinkoeff[3]:=8.333333333333e-3;          (*  1/120                 *)
56:       sinkoeff[4]:=-1.984126984127e-4;         (* -1/5040                *)
57:       sinkoeff[5]:=2.755731922399e-6;          (*  1/362880              *)
58:       sinkoeff[6]:=-2.505210838544e-8;         (* -1/39916800            *)
59:       sinkoeff[7]:=1.605904383682e-11;         (*  1/6227020.8e3         *)
60:       sinkoeff[8]:=-7.647163731820e-13;        (* -1/1.307674368e12      *)
61:     end;
62:
63:     procedure cosbelegen(var coskoeff:vektor);
64:     (* Belegen von coskoeff mit den Koeffizienten der Taylor-Reihe fuer Cosinus*)
65:     begin
66:       coskoeff[1]:=-0.5;                       (* -1/2                   *)
67:       coskoeff[2]:=4.166666666667e-2;          (*  1/24                  *)
68:       coskoeff[3]:=-1.388888888889e-3;         (* -1/720                 *)
69:       coskoeff[4]:=2.480158730159e-5;          (*  1/40320               *)
70:       coskoeff[5]:=-2.755731922399e-7;         (* -1/3628800             *)
71:       coskoeff[6]:=2.087675698787e-9;          (*  1/479001600           *)
72:       coskoeff[7]:=-1.147074559773e-11;        (* -1/8.71782912e10       *)
73:       coskoeff[8]:=4.779477332387e-14;         (*  1/2.0922789888e13     *)
74:     end;
75:
76:   begin
77:     n:=1;
78:     if x>=0.785398163397 then                  (* wenn x aus pi/4..pi/2  *)
79:     begin                                      (* dann:                  *)
80:       cosbelegen(koeff);                       (* Cosinus Reihenentwicklg*)
81:       x:=1.570796326795-x;                     (* mit x=pi/2-x           *)
82:       x2:=x*x;                                 (* x^2 einmal berechnen   *)
83:       hlp1:=x2;                                (* Variable fuer x^n      *)
84:       hlp2:=1;                                 (* Variable fuer Wert des *)
85:                                                (*   Polynoms             *)
86:     end
87:     else                                       (* wenn x aus 0..pi/4     *)
88:     begin                                      (* dann:                  *)
89:       sinbelegen(koeff);                       (* Sinus Reihenentwicklung*)
90:       x2:=x*x;                                 (* x^2 einmal berechnen   *)
91:       hlp1:=x;                                 (* Variable fuer x^n      *)
92:       hlp2:=0;                                 (* Variable fuer Wert des *)
93:                                                (*   Polynoms             *)
94:     end;
95:     hlp3[n]:=koeff[n]*hlp1;                    (* 1. Glied der Entwicklg *)
96:     while abs(hlp3[n]) >= eps do               (* Abbruchbedingung       *)
97:     begin
98:       n:=n+1;                                  (* naechstes Glied        *)
99:       hlp1:=hlp1*x2;                           (*   ermitteln            *)
100:      hlp3[n]:=koeff[n]*hlp1;                  (*                        *)
101:    end;
102:    for i:=n-1 downto 1 do                     (* Addition der einzelnen *)
103:      hlp2:=hlp2+hlp3[i];                      (* Glieder                *)
104:    sns:=hlp2;                                 (* Rueckgabe des          *)
105:  end;                                         (*   Naeherungswertes     *)
```

Listing 2: Fortsetzung

Das richtige Ergebnis lautet
1.000000500e10.
Alles klar?

```
program test1;
var i:integer;
    x:real;
begin
  x:=0;
  for i:= 1 to 10000 do
    x:=x+0.05;
  x:=x+1e10;
  writeln(x);
end.
```

```
program test2;
var i:integer;
    x:real;
begin
  x:=1e10;
  for i:=1 to 10000 do
    x:=x+0.05;
  writeln(x);
end.
```

test1/test2: Zwei kleine Programme zum Genauigkeitstest. Vergleichen Sie selbst.

Verwendung der Routinen: Will man nun diese Routinen verwenden, muß man nur am Anfang des Prozedurvereinbarungsteils die Datei mit (★$I sincos.pas★) einfügen. Der Pascalcompiler ersetzt dann selbsttätig die alten Funktionen durch die neuen.

Peter Sadoni



Joystick in Aktion

Joystickabfrage mit GfA-BASIC

Nach der Joystickroutine in Omikron-BASIC in der Juniausgabe zeigen wir Ihnen jetzt eine Variante in GfA-BASIC, um den freudigen Knüppel auch in eigenen Programmen benutzen zu können.

Wenn man nach einer Lösung sucht, probiert man es wahrscheinlich erst einmal mit dem Durchstöbern von BASIC Büchern. An irgendeiner Stelle wird dann erklärt, wie man mittels 'Peek' auf eine bestimmte Adresse zugreifen kann, die die Daten des Joysticks enthält. Doch hat diese Lösung dann meist den Nachteil, daß nur der zweite Joystick benutzt werden kann.

Es geht aber auch anders: Wir sprechen unmittelbar den Tastaturprozessor an und versuchen, die Routine möglichst allgemein zu formulieren. Wenn man beim neuen TOS die gleiche(n) absoluten Adresse(n) verwendet (siehe Lösung 1), könnte man den Knüppel noch so stark verrenken, eine Abfrage des Joysticks wäre nicht möglich. So ist die zweite Lösung besser, denn sie läuft dann auch mit dem neuen TOS. Dieses wurde ja bekanntlich zwecks Blitter aufgeräumt und somit verändert. Um nun Kompatibilität auch zur neuen Version des TOS zu erreichen, benutzen wir hier bestimmte Betriebssystemfunktionen. Ein Beispiel zum Verständnis: Xbios(2) liefert die Anfangsadresse des angezeigten Bildschirmspeichers X. Wenn sich X in einem neuen TOS ändert, liefert Xbios(2) aber dennoch die richtige Adresse. Deshalb funktionieren auch Programme, die diese Funktion verwenden, auf ST-Computern, die eine unterschiedliche Speicherkapazität haben. (Der ST schiebt die Bildschirmadresse im Speicher immer nach oben.) Für unsere Zwecke verwenden wir die Funktion Xbios($14). Diese liefert die Anfangsadresse einer Tabelle, in der wiederum Adressen von Routinen stehen, die Daten des Tastaturprozessors verarbeiten. In dieser Tabelle ist z. B. neben der Joystickadresse auch die Adresse der Maus-Routine 'eingetragen'. Um nun für beide Joysticks die Adressen zu erhalten, addieren wir zum Tabellenanfang 'Tabanf%' jeweils einen Wert. Für Joystick 1 ist dies 60, für den anderen 61. Diese Adressen werden später in der Prozedur 'Pruefe' auf Ereignisse untersucht. Nun muß der Tastaturprozessor so programmiert werden, daß er ab jetzt die Joystickwerte entgegennimmt. Dies geschieht mit der Funktion Xbios($19), welche dem Prozessor dafür den Befehlcode $14 übergibt. Omikron-BASIC ist hier eleganter. Hier hätte man einfach einen Kanal zum Tastaturprozessor mittels 'Open „K"' eröffnet, und dann in diesen die Befehlcodes geschrieben, bzw. die Joystickwerte ausgelesen. Nun aber zurück zu unserem Programm. Um die Joystickwerte zu erhalten, müssen wir die Adressen Ereignis1% und Ereignis2% mittels des BASIC Befehls 'Peek' auslesen. Die Werte ändern sich entsprechend den Bewegungen der Joysticks (siehe Ereigniswerte im Listing). Die diagonalen Werte ergeben sich logischerweise aus den anliegenden Werten. Beispiel:

rechtsoben = rechts+oben

-> 8+1 = 9.

```
' >> Joystickabfrage von Dirk Beyer <<
' >>         in GfA BASIC          <<
'
Hidem              ! Maus aus
Tabanf%=Xbios(34) ! Tabelle mit Adresse fuer Abfrage
Ereignis1%=Tabanf%+60       ! Adresse fuer Joystick 1
Ereignis2%=Tabanf%+61       ! Adresse fuer Joystick 2
'
' Befehlcodes, die durch Xbios(&h19) an den
' Tastaturprozessor geschickt werden:
'
A$=Chr$(&H14) ! Joystick abfragen
'
B$=Chr$(&H15) ! \
'                 Maus wieder an
C$=Chr$(&H8)  ! /
'
X=Xbios(&H19,3,L:Varptr(A$)) ! Joystickabfrage ein
While Inkey$=""
  Gosub Pruefe
Wend
'
X=Xbios(&H19,3,L:Varptr(B$)) ! Joystickabfrage aus
X=Xbios(&H19,3,L:Varptr(C$)) ! Maus ein
'
```

```
Procedure Pruefe
  '
  '  Ereigniswerte:
  '
  '    5   1   9
  '      \ | /
  '    4 - 0 - 8
  '      / | \
  '    6   2  10
  '
  '  Feuertaste = 128
  '  Ausserdem möglich: z.B. Feuer + links
  '                          = 128  +   4
  '                          = 132
  J1%=Peek(Ereignis1%)
  J2%=Peek(Ereignis2%)
  If J1%<>0
    Print At(22,10);"Joystick 1: ";J1%;"  "
  Endif
  If J2%<>0
    Print At(40,10);"Joystick 2: ";J2%;"  "
  Endif
Return
```

Diese Werte für links, rechts, Feuer usw. können beliebig kombiniert sein. Wird der Knüppel nach oben gedrückt, während der Feuerknopf gedrückt ist, entspricht dies dem Ereigniswert 1 + 128 = 129. Das Programm verharrt solange in einer Abfrageschleife, bis eine Taste gedrückt wird. Danach wird dem Tastaturprozessor ein Befehl geschickt, um die Maus wieder zu aktivieren.
Probieren Sie diese Joystickroutine aus! Sie können auch auf eine andere Art feststellen, welcher Knüppel bewegt wurde. Dazu fragt man die Adresse 'Tabanf + 59' ab. Enthält diese Adresse den Wert 254, wurde Joystick 1 benutzt, anderfalls(255) der zweite.

Sie können mit dem Tastaturprozessor „beliebig" experimentieren. Doch rate ich Ihnen, das Programm im Editor immer vorher abzuspeichern, bevor Sie in die Tiefen der Tastatur vordringen. Irgendwann wird der Tastaturprozessor unwillig und Sie lösen einen Systemabsturz aus...

Wir wollen nun die Maus näher untersuchen. Klicken wir einfach einmal die linke Taste, dann zeigt unser ahnungsloses Programm einen Wert für Joystick 1 an, der dem gedrückten Feuerknopf entspricht. Die rechte Taste ist dem Knopf des zweiten Joysticks gleichzusetzen. Die Daten für die Richtungen und die Tasten erscheinen als Summe. Es gibt im Vergleich zum Joystick einen Unterschied. Anstatt daß die Richtungsdaten – wie beim Joystick – nur dann vorhanden sind, wenn dieser auch wirklich betätigt wird, bleiben die Mausdaten solange erhalten, bis eine Richtungsänderung eintritt. Nun hat die Maus z. B. den Wert 11 geliefert. Jetzt ziehen wir den Mausstecker aus der Buchse. Anstatt der 11 wird jetzt eine 0 angezeigt. Nun schließen wir die Maus wieder an und stellen fest, daß in der Speicherstelle nun wieder der alte Wert 11 steht. Der Tastaturprozessor weiß also, ob eine Maus angeschlossen ist. Und er speichert die Daten. Dieses Spielchen funktioniert auch am zweiten Port. So, und nun sind Sie an der Reihe.

Dirk Beyer



Mini Clip

5 LATTICE C zu gewinnen

LATTICE C gibt es nun in der neuen Version 3.04 – und Sie können einen davon gewinnen. Schicken Sie uns dazu ein kleines Programm, das in C geschrieben wurde ein. Das Programm (Programmchen) sollte eine interessante Funktion erfüllen, die anderen Programmierern von Nutzen ist.

Gestiftet von G. KNUPE GmbH & Co. KG

Die Auswahl der Gewinner wird von einer redaktionsinternen Jury vorgenommen.

Der Rechtsweg ist ausgeschlossen. Die Teilnahme von Redaktions- und Verlagsangehörigen ist nicht zugelassen.

Einsendeschluß: 1.1.88

'MERLIN'-Computer GmbH
'LATTICE C'
Industriestr. 26
D-6236 Eschborn

30 OMIKRON.Programme zu gewinnen (Basic Compiler oder IDEAL-Assembler) nach Wahl.

Wir suchen Tips & Tricks in OMIKRON.Basic. Da bekanntlich gerade kleinere Routinen für die Anwender von Interesse sind, möchten wir hier unsere Leser aufrufen, falls sie solche zur Hand haben, uns diese zu schikken. Dabei ist es egal ob es sich um eine eigenständiges Programm, eine Prozedur oder Tips & Tricks anderer Art handelt. Wichtig ist nur, daß auch andere Leser daraus einen Nutzen ziehen können.

Gestiftet von Omikron-Software

Als Belohnung erhält jeder Einsender, dessen Tip veröffentlicht wird, einen OMIKRON.Compiler. Max. 30 Stück sind zu vergeben.

Der Rechtsweg ist ausgeschlossen. Die Teilnahme von Redaktions- und Verlagsangehörigen ist nicht zugelassen.

Einsendeschluß ist der 1.1.88

'MERLIN'-Computer GmbH
'Tips & Tricks OMIKRON.Basic'
Industriestr. 26
D-6236 Eschborn

5 GFA-Publisher zu gewinnen

Wer möchte gerne einen Desktop Publisher gewinnen? Hier bieten wir die Chance. Senden Sie dazu einfach eine kleine nützliche Routine in GFA Basic ein. Die 5 besten Einsendungen erhalten je ein GFA-Publisher. Die Auswahl der Gewinner wird von einer redaktionsinternen Jury vorgenommen.

Gestifte von GfA-Systemtechnik

Der Rechtsweg ist ausgeschlossen.

Angehörigen der Redaktion oder des Verlags ist die Teilnahme untersagt.

Einsendeschluß: 1.1.88

'Merlin'-Computer GmbH
'GFA-Publisher'
Industriestr. 26
D-6236 Eschborn

Schnell wie der Wind

Neue Form__Dial Routine

Zunächst werden wir die Arbeitsweise dieser Funktion erläutern. Soll in einem GEM-Programm eine Formular- oder Alert-Box dargestellt werden, wird vorher der dazu benötigte Bildschirmbereich durch den Aufruf der Form__Dial Routine reserviert. Äußerlich ist davon nichts zu merken, intern jedoch wird die Größe des Bereichs gespeichert.

Sobald der Programm-Nutzer seine Eingaben abgeschlossen hat und die Box nicht mehr benötigt wird, wird vom Programm erneut die Form__Dial Routine aufgerufen, diesmal um den Bildschirmbereich wieder freizugeben. Jetzt kommt es jedoch zu einer unmittelbaren Reaktion des Betriebssystems: für alle von diesem Bildschirmbereich betroffenen Fenster (auch Accessories) wird eine Nachricht ausgesandt, damit die zerstörten Bereiche wieder neu gezeichnet werden.

Diese Wiederherstellung des Bildschirms geht bei einigen Programmen nur sehr langsam voran. Wenn beispielsweise beim Textverarbeitungsprogramm 1st-Word ein komplizierter Bildschirminhalt (mit verschiedenen Schriftarten, ...) erneuert werden muß, so kann dies

In der Juni-Ausgabe der Zeitschrift wurde eine elegante File-Select-Box vorgestellt, die aber nur in eigenen Programmen verwendet werden kann. Mit dem hier vorliegenden Artikel wird gezeigt, daß es einfch ist, solche Änderungen bleibend (bis zum Reset) im Betriebssystem zu verankern, sodaß alle Programme davon profitieren. Als Beispiel wird hier die Form__Dial Routine des AES verwendet, die beschleunigt wird.

einige Sekunden dauern. Diese unnötigen Wartezeiten werden mit dem im folgenden beschriebenen Programm vermieden.

Überblick

Die Funktionsweise ist dabei die: Das Programm wird zuerst resident in den Speicher geladen, wo es dann etwa 30 KByte wegnimmt. Wird die Form__Dial Routine zum Reservieren eines Bildschirmbereichs aufgerufen, wird die AES-Routine umgangen, stattdessen wird der Bildschirmausschnitt in den, im Programm vorgesehenen Speicher geschrieben. Wenn der Bereich mit einem erneuten Aufruf der Funktion wieder freigegeben werden soll, wird er durch das Programm blitzartig wieder hergestellt. Es kommt zu keinen Meldungen des Betriebssystems an das Programm, daß ein Bildbereich zu erneuern ist.

Nun zu den Details: Das Programm besteht eigentlich aus zwei Teilen: dem Initialisierungsteil, der nur einmal durchlaufen wird, und dem eigentlichen Programm, das bei jedem Aufruf einer AES- oder VDI Routine angesprungen wird.

Der Initialisierungsteil

Im Initialisierungsteil wird zuerst (wie in allen GEM-Programmen) der notwenige Speicherbedarf berechnet. Dabei wird auch ein beim Eingeben des Programms festzulegender Betrag von Bytes berücksichtigt, der als Pufferspeicher verwendet wird, um den Bild-

```
                    Programmlisting - 1 -

; Michael Kofler, August 87

; das Programm umgeht AES-Routine Form_Dial;
; der zu reservierenden Bildschirmbereich wird im RAM zwischenspeichert
; und diesen Bereich bei der Freigabe blitzartig erneuert; dies bringt
; bei GEM-Programmen mit langsamen Bildschirmaufbau eine klare
; Beschleunigung des Programmablaufs, wenn Formulare, Alert-Boxen,...
; verwendet werden

          SECTION TEXT

INITGRAF EQU       $A000          ;Codes der Line-A Grafik Routinen
SHOWMAUS EQU       $A009
HIDEMAUS EQU       $A00A
GEMDOS   EQU       1              ;Trap-Nummer für GEMDOS-Funktionen
KEEP     EQU       $31            ;Fn.nummer   für KEEP PROCESS
BIOS     EQU       13             ;Trap-Nummer für BIOS-Funktionen
SETEXC   EQU       5              ;Fn.nummer   für SETEXEC
GEMVEC   EQU       34             ;Vektor des GEM-Traps
SCRADR   EQU       $44E           ;hier steht die Bildschirmstartadresse
PUFFLEN  EQU       25000          ;Größe des Pufferspeichers

; INITIALISIERUNGSTEIL
         MOVE.L    4(A7),A0            ;Programmbeginn
         MOVE.L    #$100,D4            ;+ Base Page
         ADD.L     12(A0),D4           ;+ Prg.Länge
         ADD.L     20(A0),D4           ;+ Datensegement-Länge
```

```
         ADD.L     28(A0),D4          ;+ Blocksegment-Länge
         ADD.L     #PUFFLEN,D4        ;+ Raum für Pufferspeicher und Stack
                                      ;D4 gibt benötigten Speicher an
         MOVE.L    #-1,-(A7)          ;GEM Exception-Vektor bestimmen
         MOVE.W    #GEMVEC,-(A7)
         MOVE.W    #SETEXC,-(A7)
         TRAP      #BIOS
         ADDQ.L    #8,A7
         MOVE.L    D0,A0              ;Test, ob Programm bereits resident
         CMP.L     #$1A2B3C4D,2(A0)   ;2. und 3. Wort der Exceptionroutine
                                      ; mit $1A2B3C4D vergleichen
         BEQ.S     ABBRUCH            ;wenn identisch ->> Programm abbrechen

         MOVE.L    #GEMNEW,-(A7)      ;GEM Exception-Vektor ändern:
         MOVE.W    #GEMVEC,-(A7)      ; zeigt jetzt auf den unten folgenden
         MOVE.W    #SETEXC,-(A7)      ; Programmteil
         TRAP      #BIOS
         ADDQ.L    #8,A7
         MOVE.L    D0,OLDGEM          ;alte GEM-Startadresse retten

         LEA       PUFFER,A0          ;Adressen des Pufferspeicher bestimmen
         MOVE.L    A0,STARTMEM        ; und in den vorgesehenen Programm-
         ADDA.L    #PUFFLEN,A0        ; variablen speichern
         MOVE.L    A0,ENDMEM

         CLR.W     -(A7)              ;Initialisierung beenden (Programm
         MOVE.L    D4,-(A7)           ; bleibt aber im Speicher resident)
         MOVE.W    #KEEP,-(A7)
```

Listing 1: Die neue Form__Dialroutine zeigt dem Blitter wo es langgeht.



```
                 Programmlisting - 2 -

        TRAP     #GEMDOS

ABBRUCH CLR.L    (A7)                  ;Programm abbrechen (Programm wird au
        CLR.W    -(A7)                 ; Speicher entfernt)
        TRAP     #GEMDOS


; Start der Routine, die künftig immer dann durchlaufen wird, wenn eine
; AES oder VDI Routine aufgerufen wird
GEMNEW  BRA.S    WEITER
        DC.L     $1A2B3C4D             ;Kennung der Routine
WEITER  CMPI.B   #$C8,D0               ;wurde AES-Funktion aufgerufen ?
        BNE.S    JUMPAES               ;nein, weiter im Betriebssystem
        MOVEM.L  D0-D4/A0-A3,-(A7)     ;Register retten
        MOVEA.L  D1,A0                 ;A0 zeigt auf AES-Parameterblock
        MOVEA.L  (A0),A1               ;A1 zeigt auf CONTRL-Feld
        CMPI.W   #51,(A1)              ;wurde Form_Dial aufgerufen ?
        BNE.S    CONTAES               ;nein, weiter in AES
        MOVEA.L  8(A0),A2              ;A2 zeigt auf INTIN-Feld, dort wird
                                       ;Funktion der Form_Dial Routine bestir
        CMPI.W   #0,(A2)               ;bei 0 GET-Programmteil aufrufen
        BEQ.S    GET
        CMPI.W   #3,(A2)               ;bei 3 PUT-Programmteil aufrufen
        BEQ.S    PUT
        BRA      RETURN                ;ansonsten Rücksprung ins Hauptprogra
                                       ; (d.h. schrumpfende Rahmen,... werden
                                       ; nicht gezeichnet)
```

```
ERROR   MOVE.W   #0,2(A3)              ;bei Fehler in GET/PUT Routinen:
        MOVE.W   #1,6(A3)              ; Maus wieder sichtbar machen,
        DC.W     SHOWMAUS              ; weiter in Betriebssystemroutinen
CONTAES MOVEM.L  (A7)+,D0-D4/A0-A3     ;Register zurück
JUMPAES MOVE.L   OLDGEM,A0             ;A0 .. Startadresse der GEM-Routinen
        JMP      (A0)                  ; dorthin springen


MAUSAUS MOVE.L   A2,-(A7)              ;A2 (zeigt auf Intin) retten
        DC.W     INITGRAF              ;Grafik initialisieren
        MOVE.L   4(A0),A3              ;A3 zeigt auf CONTRL-Array (für später
        DC.W     HIDEMAUS              ;Maus ausschalten
        MOVE.L   (A7)+,A2              ;A2 wieder zurück
        RTS


GET     ;zu reservierenden Bild-Bereich lesen und im Puffer speichern
        BSR      MAUSAUS               ;Maus abschalten
        BSR.S    CALC                  ;Adressen, Speicherbedarf... errechnen
        ADD.L    STARTMEM,D0           ; Feststellung, ob genug Speicher für
        CMP.L    ENDMEM,D0             ; den Bildausschnitt vorhanden ist
        BGT      ERROR                 ;zuwenig Speicher - daher normale AES-
                                       ; Routine verwenden
        MOVE.L   STARTMEM,A1           ;A1 zeigt auf Beginn es freien Speiche
        MOVE.L   A0,A2                 ;Bildschirmbereich in Pufferspeicher
LOOP1   MOVE.W   D1,D4                 ; schreiben; Register aus CALC bekannt
```

Listing 1: Fortsetzung

schirmausschnitt zwischenzuspeichern. Die Größe kann grundsätzlich frei gewählt werden – je mehr Bytes dazu verwendet werden dürfen, desto größere Bereiche können gesichert werden. Wenn zur Speicherung eines Bildbereiches nicht genügend Speicher vorhanden ist, wird die 'normale' AES Routine verwendet. Für die meisten Anwendungen reichen etwa 20 KByte. Die obere Grenze stellen 32 KByte dar (das reicht für den gesamten Bildschirminhalt, größer wird ein Formular sicher nie sein).

Im Initialisierungsteil wird überprüft, ob das Programm bereits im Speicher verankert wurde – in diesem Fall wird die Initialisierung abgebrochen. Es wird durch diese Abfrage sichergestellt, daß das Programm nicht mehrfach im Speicher ist.

Wenn das Programm noch nicht resident ist, dann wird mit der BIOS Funktion Set_Exception der Ansprungvektor für AES- und VDI-Routinen auf den folgenden Programmteil GEMNEW gerichtet. Die alte Startadresse wird in der Programmvariablen OLDGEM gespeichert und benötigt, wenn eine andere Routine als AES-Form_Dial aufgerufen wird.

Anschließend werden die Programmvariablen STARTMEM und ENDMEM mit der Start- und Endadresse des verfügbaren Pufferspeichers belegt – dieser Speicher beginnt nach dem Ende des Programms, seine Länge wird durch PUFFLEN angegeben. Danach erfolgt mit der GEMDOS-Funktion Keep_Proccess der Rücksprung in das Programm, aus dem das Programm gestartet wurde (gewöhnlich GEM-Desktop). Mit dieser GEMDOS-Funktion wird für das Programm der in D4 angegebene Speicherbedarf reserviert, der Rest wird an die GEM-Speicherverwaltung zurückgegeben. Das Programm bleibt im Speicher und kann jederzeit aufgerufen werden.

Das Hauptprogramm

Der zweite Programmteil beginnt bei dem Label GEMNEW. Dort wird eine Identifikationsnummer gespeichert, mit der verhindert werden soll, daß das Programm mehrfach installiert werden kann (siehe oben).

Dieser Programmteil wird nun (wegen der Veränderung des Einsprungvektors) bei jedem Aufruf einer AES- oder VDI-Routine aufgerufen. Es muß zuerst festgestellt werden, ob eine AES- oder VDI-Routine aufgerufen wurde. Dazu wird das Register D0 ausgewertet, das den Wert $C8 haben muß, wenn eine AES-Routine aufgerufen wird. Wenn dies nicht der Fall ist, dann wird (unter Verwendung der gespeicherten alten Einsprungadresse) die zuständige Routine des Betriebssystems aufgerufen. Gleiches geschieht, wenn zwar eine AES-Routine aufgerufen wurde, aber eine andere Funktiosnummer, als die von Form_Dial verwendet wird.

Handelt es sich bei dem Aufruf hingegen tatsächlich um die Form_Dial Routine, muß das Intin-Feld für AES-Routinen ausgewertet werden. Dort wird im ersten Wort angegeben, auf welche Weise die Funktion verwendet werden soll: mit 0 zum Reservieren eines Bildschirmteiles, mit 3 zum Freigeben desselben. Möglich sind auch die Werte 1 und 2 – so wird die Funktion dazu verwendet, schrumpfende oder wachsende Rechtecke zu zeichnen, damit es aussieht, als kämen die Boxen,... aus der Unendlichkeit oder würden eben dorthin wieder verschwinden.

In den ersten beiden Fällen werden die, bei den Labeln GET und PUT beginnenden Programmteile aufgerufen, die den Bildschirm speichern oder wiederherstellen sollen. In den beiden anderen Fällen erfolgt der Rücksprung ins Programm, von dem die Form_Dial Routine aufgerufen wurde. Es werden also keine schrumpfenden,... Rechtecke gezeichnet. Wenn jemand darauf Wert legt, muß lediglich die Anweisung BRA RETURN durch BRA CONTAES ersetzt werden – es erfolgt in diesen Fällen der Aufruf der normalen Form_Dial Routine des Betriebssystems.

Der Programmteil GET

Statt einen Bildbereich zu reservieren, wird er in diesem Programmteil im RAM zwischengespeichert. Dazu muß zuerst die Maus abgeschaltet werden, damit diese (falls sie bewegt wird), keine Störungen verursacht. Das dazu verwendete Unterprogramm benutzt die Line-A Grafik-Befehle. Es wird darin auch die Adresse des Control-Feldes dieser Befehle ermittelt und in A3 gespeichert – die Adresse wird später benötigt, wenn die Maus wieder eingeschaltet werden soll.

```
                    Programmlisting - 3 -

LOOP2    MOVE.W    (A0)+,(A1)+       ;innere Schleife für eine Bildschirm-
         DBF       D4,LOOP2          ; Zeile, deren Länge in D4 bestimmt ist
         ADD.L     #80,A2
         MOVE.L    A2,A0
         DBF       D2,LOOP1          ;äußere Schleife für Zahl der Zeilen(D2

         MOVE.L    STARTMEM,(A1)+    ;Startadresse des Blocks speichern
         MOVE.L    D3,(A1)+          ;zuletzt Kennnummer speichern
         MOVE.L    D0,STARTMEM       ;neuer Start

         BRA.S     MAUSEIN           ;Maus wieder sichtbar, Rücksprung


PUT      ;Bild-Bereich aus Puffer auf den Bildschirm schreiben
         BSR       MAUSAUS           ;Maus abschalten
         BSR.S     CALC              ;Adressen... ausrechnen
         MOVE.L    STARTMEM,A1
         CMP.L     -4(A1),D3         ;Kennnummer vergleichen
         BNE       ERROR             ;nicht passend, daher weiter im AES
         MOVE.L    -8(A1),A1         ;in A1: Startadresse des Blocks

         MOVE.L    A0,A2             ;Bildschirmausschnitt vom Puffer-
LOOP3    MOVE.W    D1,D3             ; speicher auf Bildschirm übertragen
LOOP4    MOVE.W    (A1)+,(A0)+       ;innere Schleife für eine Bildschirm-
         DBF       D3,LOOP4          ; Zeile
         ADD.L     #80,A2
```

```
         MOVE.L    A2,A0
         DBF       D2,LOOP3          ;äußere Schleife für Zahl der Zeilen
         SUB.L     D0,STARTMEM       ;Start des freien Speicher reduzieren

MAUSEIN  MOVE.W    #0,2(A3)          ;Maus wieder sichtbar machen
         MOVE.W    #1,6(A3)          ; A3 zeigt auf Grafik-Parameter
         DC.W      SHOWMAUS
RETURN   MOVEM.L   (A7)+,D0-D4/A0-A3 ;Register zurück, danach Rücksprung ins
         RTE                         ;Programm, aus dem Aufruf erfolgte


CALC     ;Koordinaten auslesen, Wortgrenzen, Speicherbedarf
         ;gegeben: A2: zeigt auf INTIN Feld
         ;gesucht: D0: Speicherbedarf in Byte     D3: Kenn-Nummer
         ;         D1: Breite in Worten           D2: Höhe in Zeilen
         ;         A0: Startadresse des Rechtecks am Bildschirm
         LEA       10(A2),A0         ;A0 zeigt auf Beginn der Parameterliste
         CLR.L     D0                ; im Intin-Feld
         MOVE.W    (A0),D0           ;D0: X1-Koordinate in Punkten
         MOVE.W    D0,D1
         ADD.W     4(A0),D1          ;D1: X2-Koordinate (X1+Breite)in Punkten
         ASR.W     #4,D0             ;D0: X1 in Worten (Division durch 16)
         ASR.W     #4,D1             ;D1: X2 in Worten
         SUB.W     D0,D1             ; --> D1: Breite minus 1 in Worten

         MOVE.W    6(A0),D2          ; --> D2: Y-Breite
```

Listing 1: Fortsetzung

Im Unterprogramm CALC werden dann aus den Daten im Intin-Block der AES-Funktion die Breite des zu rettenden Bildschirmbereiches in Worten (D1), die Zahl der Zeilen dieses Bereiches (D2), der Speicherbedarf zur Speicherung des Rechtecks (D3), die Startadresse des Bereiches am Bildschirm (A0), sowie eine Kenn-Nummer (D0) errechnet.

Der Grund, weshalb die Breite des Rechteckes in Worten angegeben wird, ist leicht zu verstehen: es wird nicht genau das angegebene Rechteck gespeichert, sondern aus Gründen der Einfachheit und Geschwindigkeit ein (meistens) etwas größerer Bereich mit Wort-Grenzen. Andernfalls hätten einzelne Bits am Rand separat berücksichtigt werden müssen.

Der Speicherbedarf (in Byte) ergibt sich grundsätzlich aus der Multiplikation Breite in Worten mal Zeilenanzahl mal zwei. Es muß aber berücksichtigt werden, daß sowohl die Zeilenanzahl als auch die Breite um eins reduzierte Werte sind, und daß außer der eigentlichen Bildinformation auch noch die eine Adresse, sowie die Kenn-Nummer gespeichert werden muß.

Die Kenn-Nummer ist notwendig, damit der PUT-Programmteil feststellen kann, ob der freizugebende Bildschirmbereich mit dem gespeicherten Bereich identisch ist (dies ist dann *nicht* der Fall, wenn nicht ausreichend Speicher vorhanden war, um das Rechteck zu speichern).

Nachdem alle diese Werte ermittelt wurden, sollte man feststellen, ob genug Speicher vorhanden ist, um den Bereich zu speichern. Wenn dies der Fall ist, dann wird in einer einfachen Schleife der Bildschirminhalt wortweise in den Pufferspeicher übertragen. Außerdem wird noch die Kenn-Nummer und die Startadresse des gerade gespeicherten Blocks im Puffer abgelegt – beide Informationen werden vom Programmteil PUT benötigt. Auch die Startadresse des Pufferspeichers muß erhöht werden, damit es möglich ist, auch mehrere (kleinere) Boxen zugleich zwischenzuspeichern (dies ist möglich, kommt in der Praxis aber ziemlich selten vor).

Anschließend verzweigt das Programm zum Label MAUSEIN, wo die Maus wieder sichtbar gemacht wird, und die Rückkehr in das Programm erfolgt, von dem die Form_Dial Routine aufgerufen wurde. Dabei ist zu beachten, daß der Rücksprung über RTE und nicht RTS erfolgt, weil Betriebssystem-Routinen beim ST über Traps aufgerufen werden und sich der Prozessor dann im 'Exception-Modus' befindet.

Wenn der Speicher nicht ausreicht, erfolgt über den Programmteil ERROR der Aufruf der Form_Dial-Routine des Betriebssystems, wozu die Maus wieder sichtbar gemacht werden muß.

Der Programmteil PUT

Dieser Programmteil wird aufgerufen, um einen Bildschirmbereich wieder frei zu geben. Wie bei GET wird zuerst die Maus unsichtbar gemacht und das Unterprogramm CALC aufgerufen. Sodann wird festgestellt, ob die errechnete Kenn-Nummer mit dem letzten Wert im Pufferspeicher identisch ist – wenn dies nicht der Fall ist, wird (über ERROR) die Form_Dial Routine des Betriebssystems aufgerufen.

Andernfalls wird aus dem Pufferspeicher die Startadresse des Blocks gelesen, in einer ähnlichen Schleife wie bei GET wird der Bildschirminhalt wieder erneuert. Die Startadresse des Pufferspeichers wird verkleinert – der Speicher kann somit wieder verwendet werden. Anschließend erfolgt, wie bei GET der Rücksprung in das Programm, welches die Form_Dial Routine verwendet hat.

Anwendung des Programmes

Das Programm läuft in dieser Form nur in der höchsten Auflösungsstufe des ST. Es kann jedoch an die anderen Modi angepaßt werden, wozu im Unterprogramm CALC die Berechnung der Breite des Rechtecks in Worten geändert werden muß (in Mid-Res: die beiden Rotierbefehle ASR mit #3 statt #4, in Low-Res mit #2).

Auch die Berechnung der Bildschirm-Startadresse muß ein wenig verändert werden: die Multiplikation muß sowohl in Low-Res, als auch in Mid-Res mit #160 statt #80 erfolgen.

Abgesehen davon läuft das Programm ohne Einschränkung mit allen anderen

```
                 Programmlisting - 4 -

MOVE.L   SCRADR,D3          ;D3: Bildschirmstartadresse
ADD.W    D0,D0              ;D0: X1 in Byte
ADD.L    D0,D3              ;zur Startadr. addieren
MOVE.W   2(A0),D0           ;D0: Y1 in Zeilen
MULU     #80,D0             ;mal 80 .. in Byte
ADD.L    D0,D3              ;zur Startadr. addieren
MOVE.L   D3,-(A7)           ;am Stack zwischenspeichern

MOVE.W   D1,D0              ;D0: Speicherbedarf in Wörtern
ADDQ.W   #1,D0              ;aus D1: Breite in Worten
MULU     6(A0),D0           ;mit Höhe (Zeilenanzahl) multipilzieren
ADD.W    D1,D0              ;nochmal Breite in Worten addieren
ADD.W    #5,D0              ;plus 5 Worte für Kenn-Nummer,...
ADD.W    D0,D0              ;mal 2 --> D0:Speicherbedarf in Byte
```

```
         MOVE.L   (A0),D3          :Kenn-Nummer bestimmen
         EOR.W    D0,D3            ; --> D3: Kenn-Nummer
         MOVE.L   (A7)+,A0         ;Startadresse vom Stack in --> A0
         RTS


;Programmvariablen

STARTMEM DS.L     1           ;Beginn des Pufferspeichers
ENDMEM   DS.L     1           ;Ende des Pufferspeichers
OLDGEM   DS.L     1           ;Zeiger auf gewöhnliche GEM-Routinen
PUFFER   DS.W     1           ;hier beginnt der Pufferspeicher, dessen
                              ;durch PUFFLEN festgelegt ist

         END
```

Listing 1: Ende

Programmen, die die Form_Dial Routine vorschriftsmäßig verwenden. Es gibt allerdings einige Programme, bei denen eine ähnliche Pufferung des Bildinhaltes schon von sich aus durchgeführt wird (z. B. GfA-BASIC bei Alert-Boxen, Tempus-Editor bei kleinen Boxen). Bei diesen Programmen kann daher keine Beschleunigung erreicht werden.

Accessories, die nur aus einem Formular bestehen, arbeiten gewöhnlich problemlos mit dem Programm zusammen. Keinen Effekt hat das Programm hingegen auf Gem-Desktop. Dort wird die Form_Dial Routine anscheinend nicht verwendet, oder der Einsprung erfolgt nicht über den GEM-Vektor. Auch die GEM-File-Select Box, die ebenfalls einen Bildschirmausschnitt reserviert und später wieder frei gibt, nützt die Form_Dial Routine entweder gar nicht, oder umgeht zumindest den GEM-Vektor.

Grundsätzlich kann nach dem hier demonstrierten Muster jede AES- oder VDI-Funktion durch eigene, schnellere oder vielseitigere Routinen ersetzt werden. Neben der Initialisierung, bei der kaum Änderungen notwendig sind, muß darauf geachtet werden, daß die alte Startadresse der Betriebssystem-Routinen gespeichert wird, daß dorthin mit JMP gesprungen wird, wenn eine andere Routine, als die zu ersetzende aufgerufen wurde, und daß ansonsten der Rücksprung ins Programm, aus dem der Aufruf kommt, mit RTE erfolgt. Es können auch mehrere solche Programme zugleich im Speicher sein. Nur sollte darauf geachtet werden, daß diese möglichst rasch wieder verlassen werden, wenn eine andere Funktion aufgerufen wird – sonst wird das gesamte Betriebssystem noch langsamer.

Michael Kofler



Wie von selbst

Auto-Ramdisk-Copy in Omikron-Basic

Zuerst also zu den File-Namen, die in der sequentiellen Datei namens „AUTO_RDC.BAT“ auf Disk gespeichert werden sollen: (Siehe Abb. 1)

Dieses Progrämmchen habe ich der Einfachheit halber an Auto-Ramdisk-Copy angehängt. Sie tragen also statt meiner File-Namen die von Ihnen gewünschten ein, ändern bei Bedarf den Ramdisk-Buchstaben und starten den Batch-Maker mit RUN BATCH_MAKER. Übrigens können Sie diese Datei jederzeit ändern, indem Sie den Batch-Maker mit den neuen File-Namen versehen und dann starten.

Auf eine Erklärung des Listings kann wegen des reichlichen Gebrauchs von dokumentierenden Rems sicher verzichtet werden. Gestartet wird AUTO_RDC.BAS entweder, was am bequemsten ist, mit Doppelklick direkt vom Desktop (nachdem die Ramdisk eingerichtet wurde), oder mit RUN „AUTO_RDC.BAS“ unter Omikron-Basic.

Wenn alle Batch-Files fehlerfrei auf das Ziel-Laufwerk kopiert wurden, löscht

Bei einer Speichergröße von einem Megabyte, wie sie der Atari ST bietet, gehört eine Ramdisk bei den meisten ST-Usern zum Desktop wie der Papierkorb. Da nicht jede Ramdisk ein Auto-Copy-Programm zur Verfügung stellt, das automatisch vom User ausgesuchte Files in die Ramdisk kopiert, mußte auch ich diese Files erst markieren und dann in die Ramdisk schieben, um sie so zu kopieren: Eine mühselige, fehlerträchtige und zeitraubende Arbeitsweise angesichts der Leistungsfähigkeit des Atari ST. Dabei muß ein solches Auto-Copy-Programm eigentlich gar nicht viel können: Es soll die File-Namen der zu kopierenden Dateien von Diskette lesen, um diese anschließend von Disk A auf Disk C (oder D etc.) zu kopieren. Um dieses Programm universell einsetzen zu können, soll sowohl der Buchstabe des Quell- als auch der des Ziel-Laufwerkes eingelesen werden.

sich Auto-Ramdisk-Copy selbst, da ein nochmaliger Start auf jeden Fall einen Fehler verursachen würde.

Sollte Ihre Ramdisk im Icon einen anderen Buchstaben als C aufweisen, so müssen Sie im BATCH_MAKER den Buchstaben eintragen, den Ihre Ramdisk im Icon hat.

Es sei noch erwähnt, daß Auto-Ramdisk-Copy durch die Tatsache, daß sowohl das Quell- als auch das Ziel-Laufwerk aus dem Batch-Job eingelesen werden, auch als gewöhnliches Kopierprogramm benutzt werden kann (z. B. Quell-Laufwerk: A, Ziel-Laufwerk: B und der Batch-Job natürlich auf dem Quell-Laufwerk).

Der Zeitaufwand, den das Abtippen der 78 Zeilen erfordert, lohnt sich sehr schnell, da durch Auto-Ramdisk-Copy das leidige Selektieren und Hin- und Hergeschiebe entfällt.

Also ab heute schneller, sicherer und bequemer arbeiten mit Auto-Ramdisk-Copy!

PANCREATION

```
0 '****************************************
1 '
2 '    A U T O - R A M D I S K - C O P Y
3 '
4 '****************************************
5 '
6 '    Ein ebenso kleines
7 '    wie nützliches Hilfsprogramm
8 '    für alle
9 '    Omikron-Basic-Programmierer
10 '
11 '      PANCREATION 1987
12 '
13 '     Rainer Pankratz
14 '     Im Weidenkamp 18
15 '     3340 Wolfenbüttel
16 '
17 '****************************************
18 '
19 CLS '                              Bildschirm löschen
20 ON ERROR GOTO Fehler'              Fehler abfangen
21 DIM Batch$(64)'                    Platz für Batch-Liste
22 PRINT CHR$(27);"f"'                Cursor ausschalten
23 PRINT : PRINT
24 PRINT TAB (12);"**********************************************************"
25 PRINT TAB (12);"*                                                        *"
26 PRINT TAB (12);"*            A U T O - R A M D I S K - C O P Y           *"
27 PRINT TAB (12);"*                                                        *"
28 PRINT TAB (12);"*                    PANCREATION '87                     *"
29 PRINT TAB (12);"*                                                        *"
30 PRINT TAB (12);"**********************************************************"
31 PRINT
32 '
33 OPEN "I",1,"AUTO_RDC.DAT"'        Batch-Datei öffnen
34 Count=0'                           Zähler auf Null setzen
35 INPUT #1,Quell$'                   Quell-Laufwerk lesen
36 INPUT #1,Ziel$'                    Ziel-Laufwerk lesen
37 WHILE NOT EOF(1)'                  Solange noch Daten vorliegen ...
38    Count=Count+1'                  Zähler erhöhen
39    INPUT #1,Batch$(Count)'         File-Name speichern
40 WEND '                             bis keiner mehr vorhanden
41 CLOSE 1'                           Batch-Datei schließen
42 IF Count=0 THEN ERROR 1'           bei Null Fehler erzeugen
43 REPEAT '                           Kopierschleife
44    Loop_Nr=Loop_Nr+1'                   Zähler erhöhen
45    PRINT TAB (23);Batch$(Loop_Nr);" ";' File-Namen anzeigen
46    PRINT TAB (41);">>>>";'              Vorgang anzeigen
47    COPY Quell$+":\"+Batch$(Loop_Nr) TO Ziel$+":\"'  Kopieren
48    PRINT TAB (41);"    ";'              Anzeige löschen
49    PRINT TAB (48);"  [ OK ]": PRINT '   hat geklappt
50 UNTIL Loop_Nr=Count'               bis alle kopiert sind
51 CLS '                              Bildschirm löschen
52 PRINT ">>> ";Ziel$;":": PRINT '    Ramdiskinhalt
53 FILES Ziel$+":\*.*"'               anzeigen
54 NEW '                              AUTO-RAMDISK-COPY löschen
55 '                                  Fertig!
56-Fehler'                            Fehlerbehandlung
57 MOUSEON '                          Maus ein und Alert zeigen
58 FORM_ALERT (1,"[1][Es ist ein Fehler aufgetreten||Bitte Diskinhalt
en!][ OK ]",Button)                         prüfen|und nochmal versuch
59 MOUSEOFF '                         Maus aus
60 CLS '                              Bildschirm löschen
61 PRINT ">>> "+Quell$;":": PRINT '   Quelldisk Inhalt zeigen
62 FILES Quell$+":\*.*"'              weil wahrscheinlich Datei
63 RESUME Ende'                       nicht vorhanden
64-Ende
65 END '                              Fertig!
66 '
67 '*** BATCH-MAKER *** '              Batch-Datei erstellen
68-Batch_Maker
```




```
69 OPEN "O",1,"AUTO_RDC.BAT"'        Datei öffnen
70 '
71 PRINT #1,"A"'                     Quell-Laufwerk schreiben
72 PRINT #1,"C"'                     Ziel-Laufwerk schreiben
73 PRINT #1,"FILE_NR1.XYZ"'          die zu
```

```
74 PRINT #1,"FILE_NR2.XYZ"'          kopierenden File-Namen
75 PRINT #1,"FILE_NR3.XYZ"'          schreiben
76 '                                 Liste ist erweiterungsfähig!
77 CLOSE 1'                          Datei schließen
78 END '                             Fertig!
```


```
67 '*** BATCH-MAKER *** '             Batch-Datei erstellen
68-Batch_Maker
69 OPEN "O",1,"AUTO_RDC.BAT"'         Datei öffnen
70 '
71 PRINT #1,"A"'                      Quell-Laufwerk schreiben
72 PRINT #1,"C"'                      Ziel-Laufwerk schreiben
73 PRINT #1,"FILE_NR1.XYZ"'           die zu
74 PRINT #1,"FILE_NR2.XYZ"'           kopierenden File-Namen
75 PRINT #1,"FILE_NR3.XYZ"'           schreiben
76 '                                  Liste ist erweiterungsfähig!
77 CLOSE 1'                           Datei schließen
78 END '                              Fertig!
```



Abbildung 1: Das Batch File mit den zu kopierenden Files

Mini Clip

Wie schon auf der Titelseite erwähnt, wird diese Rubrik zu einem ständigen und wichtigen Bestandteil dieser Zeitschrift. Gedacht ist sie für all diejenigen, die ihren Rechner selbst programmieren, und dazu Tips und Anregung gebrauchen können. Diese Rubrik kann aber nur dann bestehen, solange viele Leser (Sie eingeschlossen) sich daran beteiligen. Wir fordern Sie deshalb auf, Ihre Ideen in einen Umschlag zu stecken und auf dem schnellsten Weg zu uns zu schicken. Voraussetzungen gibt es praktisch keine: es muß lediglich interessant sein und so dokumentiert, daß auch andere User daraus einen Nutzen ziehen können. Auch bei den Sprachen gibt es keinerlei Einschränkungen: PASCAL, C, BASIC, Assembler, Modula-2, APL, Fortran, Forth, LISP, Prolog, PEARL, ... und sogar LOGO ist erwünscht.

Veröffentlichte Programme werden natürlich angemessen honoriert.

Bitte beachten Sie bei der Einsendung:

Schicken Sie den Quelltext und das evtl. compilierte Programm, sowie die Dokumentation auf Papier und Diskette.
Die Diskette wird zurückerstattet.

Einsendungen direkt an die Redaktion:

MERLIN-Computer GmbH
ST-Computer Redaktion
'Programmier Praxis'
Industriestr. 26
D-6236 Eschborn
Tel: 0 61 96 / 48 18 14

Dem Klicken ein Ende

Hoch- und Tiefsetzen auf Tastendruck

WORDPLUS ist eine leistungsfähige Textverarbeitung, allerdings mit einigen zum Teil erheblichen Mängeln. Eines dieser Probleme wird mit dem folgenden Programm beseitigt: Das Hoch- und Tiefstellen kann ab sofort über einen einzigen Tastendruck erreicht werden.

Es waren allerdings gewaltige maschinensprachliche Verrenkungen notwendig, bis das angestrebte Ziel erreicht werden konnte. Dabei ist das Prinzip relativ einfach: Ein kleines speicherresidentes Assemblerprogramm fängt alle GEM-Calls der Textverarbeitung ab und prüft, ob innerhalb einer EVNT_MULTI-Warteschleife die Tastatur abgefragt werden soll. Wenn ja, so wird anschließend überprüft, ob eine der Tasten HELP und UNDO betätigt wurde. Ist dies der Fall, so schreibt das Programm eine 'gefälschte' Meldung in den Ereignispuffer, der WordPlus glauben macht, der Anwender hätte einen Menüeintrag angeklickt. Das ist im Prinzip alles.

Kommen wir zu einer genaueren Beschreibung des Assembler-Programms: Die Speicherbedarfsberechnung am Beginn dürfte mittlerweile allseits bekannt sein. Interessant wird es in den Zeilen 22 bis 33: Hier schmuggelt sich das Programm in den ansonsten nicht benutzten TRAP #3-Vektor ein und liest den TRAP #2-Vektor aus, der normalerweise für alle GEM-Calls zuständig ist – schließlich muß das Programm ja wissen, wo der 'übliche' GEM-Einsprung liegt.

Der Kern des Programms beginnt natürlich beim Label *gemvektor*, daß bei jedem TRAP #3 automatisch angesprungen wird. Hier muß zunächst überprüft werden, ob ein AES- oder VDI-Aufruf stattfand, was man aus dem Register d0 entnehmen kann. Im Falle eines AES-Calls steht dann in a0 ein Zeiger auf dem sogenannten AES-Parameterblock, der eine Tabelle auf die vom AES benutzten Arrays enthält. Mit deren Hilfe kann man nun die Art des Aufrufs feststellen.

Wenn nun AES aufgerufen werden soll, so legt man einfach die Rücksprungadresse und den Prozessor-Status(!) auf den Stapel, und springt zu der vorher erfragten Einsprungadresse. AES beendet mit einem *rte* (Return from exeption). Der Rest des Programms dürfte durch die Kommentare im Listing relativ leicht zu durchschauen sein. Wichtig ist natürlich, daß auch unser Programm mit einem rte endet, da es ja durch einen TRAP #3 aufgerufen wird.

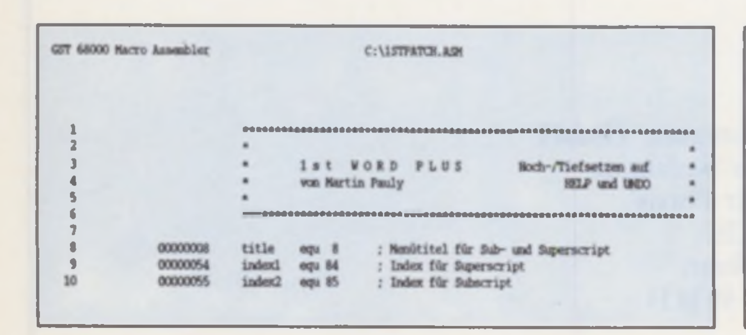
Nun wird sich manch einer fragen: Woher weiß denn WordPlus überhaupt, daß es das AES über TRAP #3 und nicht – wie gewöhnlich – über einen TRAP #2 ruft? Ganz einfach: dazu wird WordPlus ganz gepatched. Benutzen Sie dazu bitte das kurze GFA-BASIC-Programm, daß diesen Patch an der Version 1.89 (und nur an dieser!) automatisch vornimmt. Sollte es in Zukunft eine neue Version von WordPlus geben, so werde ich selbstverständlich auch dazu ein Patch-Programm erstellen, aber vielleicht kommen die Autoren dieses Programms bis dahin vielleicht selbst auf die Idee, daß unter den ST-Anwendern nicht nur absolute Maus-Fanatiker sind.

Noch eine Frage: Warum leiten wir nicht den TRAP #2 selber auf unsere Routine, was doch den Patch ersparen würde? Ganz einfach: Weil es nämlich nicht geht. Das heißt, es geht schon, nur stürzt Ihr Rechner dann ab, sobald sich außer WordPlus auch noch Accessories im Speicher befinden. Bitte fragen Sie mich jetzt nicht, warum dem so ist – ich weiß es nämlich selber nicht, und sollte ein Leser wissen, warum das so ist, so bitte ich diesen, mir das Geheimnis zu verraten, das mich schlaflose Nächte gekostet hat. So ist das Problem jedenfalls nicht sehr elegant gelöst, aber es funktioniert.

Es bleibt noch zu erwähnen, daß 1STPATCH.PRG unbedingt vor dem ersten Laden des gepatchten WordPlus gestartet werden muß, da ja sonst der TRAP #3 die schönsten Bomben, aber leider keine GEM-Calls zur Folge hat. Übrigens „Hochstellen" können Sie über HELP, „Tiefstellen" mit UNDO ein- und auschalten.

Das war's – ich wünsche Ihnen viel Spaß beim Abtippen; den Mathe-Hausaufgaben auf dem ST steht nun nichts mehr im Wege!

Martin Pauly



```
GST 68000 Macro Assembler                        C:\1STPATCH.ASM

 1                 ************************************************************
 2                 *                                                          *
 3                 *      1 s t   W O R D   P L U S      Hoch-/Tiefsetzen auf *
 4                 *      von Martin Pauly                       HELP und UNDO *
 5                 *                                                          *
 6                 ************************************************************
 7
 8       00000008  title   equ 8     ; Menütitel für Sub- und Superscript
 9       00000054  index1  equ 84    ; Index für Superscript
10       00000055  index2  equ 85    ; Index für Subscript
```

```
11
12
13 01 00000000                   section code      ; DRI: .text
14                               opt abs           ; DRI: entfällt
15
16 01 00000000 206F0004          move.l 4(sp),a0   ; Die üblichen Berechnungen...
17 01 00000004 2C3C00000100      move.l #$100,d6
18 01 0000000A DCA8000C          add.l 12(a0),d6
19 01 0000000E DCA80014          add.l 20(a0),d6
20 01 00000012 DCA8001C          add.l 28(a0),d6
21
22 01 00000016 2F3C0000004A      move.l #gemvektor,-(sp)
23 01 0000001C 3F3C0023          move.w #35,-(sp)
24 01 00000020 3F3C0005          move.w #5,-(sp)
25 01 00000024 4E4D              trap #13          ; Einsprung über TRAP #3 initialisieren
```

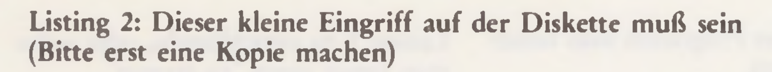
Listing 1: Assembler Listing zum Hoch- und Tiefsetzen.

```
26 01 00000026 508F             addq.l #8,sp
27
28 01 00000028 2F3CFFFFFFFF     move.l #-1,-(sp)
29 01 0000002E 3F3C0022         move.w #34,-(sp)
30 01 00000032 3F3C0005         move.w #5,-(sp)
31 01 00000036 4E4D             trap #13          ; 'Alten' GEM-Vektor (TRAP #2) merken
32 01 00000038 508F             addq.l #8,sp
33 01 0000003A 23C000000000     move.l d0,oldvektor
34
35 01 00000040 4267             clr.w -(sp)
36 01 00000042 2F06             move.l d6,-(sp)
37 01 00000044 3F3C0031         move.w #$31,-(sp)
38 01 00000048 4E41             trap #1           ; ...aber resident halten (KEEP PROCESS)
39
40                         .
41
42 01 0000004A             gemvektor:
43 01 0000004A 0C4000C8         cmpi.w #$c8,d0    ; AES-Aufruf oder 'nur' VDI?
44 01 0000004E 6618             bne.s normal      ; Nicht um's VDI kümmern!
45
46 01 00000050 2041             move.l d1,a0      ; Zeiger auf AES-Parameterblock
47 01 00000052 23C800000004     move.l a0,adsave  ; retten
48 01 00000058 2250             move.l (a0),a1    ; a1 enthält jetzt Pointer auf Contrl-Array
49
50 01 0000005A 0C510019         cmpi.w #25,(a1)   ; EVNT_MULTI Aufruf der Applikation? (Opcode)
51 01 0000005E 6608             bne.s normal      ; Nein, dann normal weiter!
52
GST 68000 Macro Assembler                         C:\1STPATCH.ASM


53 01 00000060 487900000070     pea return        ; Rücksprungadresse und
54 01 00000066 40E7             move.w sr,-(sp)   ; Statusregister auf den Stack werfen
55
56 01 00000068             normal:
57 01 00000068 207900000000     move.l oldvektor,a0
58 01 0000006E 4ED0             jmp (a0)          ; normalen TRAP #2 ausführen
59
60 01 00000070             return:
61 01 00000070 207900000004     move.l adsave,a0
62 01 00000076 2268000C         move.l 12(a0),a1  ; GEM Intout-Array
63 01 0000007A 24680010         move.l 16(a0),a2  ; GEM Addrin-Array
64 01 0000007E 3011             move.w (a1),d0    ; Ereignis-Bitfeld von EVNT_MULTI
65 01 00000080 02400001         andi.w #1,d0      ; Taste gedrückt?
66 01 00000084 673A             beq.s ende        ; Nein? Macht nichts.
67 01 00000086 0C696200000A     cmpi.w #$6200,10(a1)   ; Scancode der HELP-Taste im Hi-Byte
68 01 0000008C 6616             bne.s test2
69 01 0000008E 2452             move.l (a2),a2    ; Superscript
70 01 00000090 32BC0010         move.w #16,(a1)   ; EVNT_MESAG 'einschmuggeln'
71 01 00000094 34FC000A         move.w #10,(a2)+  ; Menu_Selected
72 01 00000098 429A             clr.l (a2)+       ; Filler
73 01 0000009A 34FC0008         move.w #title,(a2)+    ; Menü_Titel
74 01 0000009E 34BC0054         move.w #index1,(a2)    ; Menü_Index
75 01 000000A2 601C             bra.s ende
76 01 000000A4             test2:
77 01 000000A4 0C696100000A     cmpi.w #$6100,10(a1)   ; Oder vielleicht UNDO gedrückt?
78 01 000000AA 6614             bne.s ende
79 01 000000AC 2452             move.l (a2),a2
80 01 000000AE 32BC0010         move.w #16,(a1)   ; s.o.
81 01 000000B2 34FC000A         move.w #10,(a2)+
82 01 000000B6 429A             clr.l (a2)+
83 01 000000B8 34FC0008         move.w #title,(a2)+
84 01 000000BC 34BC0055         move.w #index2,(a2)
85
86 01 000000C0             ende:
87 01 000000C0 4E73             rte
88
89 02 00000000                  section storage   ; DRI: .bss
90 02 00000000             oldvektor:
91 02 00000000 00000004         ds.l 1
92 02 00000004             adsave:
93 02 00000004 00000004         ds.l 1
94
95                              end
```

Listing 1: Fortsetzung

```
' Patch für WORDPLUS.PRG zu 1STPATCH
' von Martin Pauly
' ST-Computer 11/87
'
' Erst Backup dann Patch
'
Open "U",#1,"WORDPLUS.PRG"
Seek #1,141355
Out #1,67
Close #1
```



Listing 2: Dieser kleine Eingriff auf der Diskette muß sein (Bitte erst eine Kopie machen)



Mini Clip

ST-Basic lebt wieder

Wie schon im vorletzten Heft beschrieben, wurde das ST-Basic stark überarbeitet. Sämtliche Fehler, die das Programm in die Versenkung verschwinden ließen, wurden behoben. Weiterhin wurde das ST-Basic insgesamt wesentlich schneller und viele Befehle kamen hinzu.

Ausgewählte Kapitel der Basic-Programmierung

Teil 3

Strukturen und Prozeduren

Dieses Kapitel unserer zweimonatlichen Serie, stellt eines der wichtigsten im modernen Basic dar. Die Betonung liegt hierbei auf modern, denn ältere Basic Versionen haben davon wenig bis gar nichts zu bieten.

Struktur, was ist das eigentlich?

Im alltäglichen Leben ist dieser Begriff klar; eine Landschaftsstruktur beispielsweise stellt den Aufbau und die Gestalt der Landschaft dar. Beim Programmieren ist dies nicht viel anders. Auch ein Programm hat eine äußere Form, ein Gerüst, ein Aussehen. Dem Anwender eines Programmes mag dies völlig egal sein, Hauptsache es funktioniert. Der Programmierer hingegen denkt darüber anders. Ihm bedeutet die Form seines Programmes sehr viel, denn er muß es verstehen können. Dies verdeutlicht Beispiel 1 und 2. Man muß deutlich darauf hinweisen, daß beide Listings die gleiche Wirkung haben, aber schauen Sie doch einmal welches Programm man besser verstehen kann.

```
100 'OMIKRON.Basic
110 'Spagetti Version
120 '
130 Kontostand=3000
140-Start
150 CLS
160 PRINT "Guten Tag"
170 PRINT "Ihr Kontostand beträgt ";Kontostand
180 INPUT "Wieviel Geld möchten Sie abheben ?";Geld
190 IF Kontostand-Geld<=0 THEN GOTO Nix_Abheben
200 FOR I=Kontostand TO Kontostand-Geld STEP -1
210 PRINT @(20,20);I;"    ": NEXT I
220 Kontostand=Kontostand-Geld: GOTO Ende
230-Nix_Abheben
240 PRINT "Soviel Geld haben Sie nicht"
250-Ende
260 INPUT "Wollen Sie noch etwas abheben ?";Antwort$
270 IF Antwort$="n" OR Antwort$="N" THEN
280 PRINT "Auf Wiedersehen": END ELSE GOTO Start ENDIF
290
```

Listing 1: Der berüchtigte Spaghetticode

Versuchen Sie bitte, bevor Sie die nächsten Zeilen lesen, den Sinn von Listing 1 zu entschlüsseln, oder sollte man besser sagen 'zu erraten'.

```
100 ' OMIKRON.Basic
110 '
120 Kontostand=3000
130 '
140 REPEAT
150   CLS
160   PRINT "Guten Tag"
170   PRINT "Ihr Kontostand beträgt ";Kontostand
180   INPUT "Wieviel möchten Sie abheben ?";Geld
190   IF Kontostand-Geld>=0 THEN
200    FOR I=Kontostand TO Kontostand-Geld STEP -1
210      PRINT @(20,20);I;"  "
220    NEXT I
230    Kontostand=Kontostand-Geld
240   ELSE
250     PRINT " Soviel Geld haben Sie nicht mehr !"
260   ENDIF
270   INPUT "Wollen Sie noch etwas abheben ?  [j:n]";Antwort$
280 UNTIL Antwort$="n" OR Antwort$="N"
290 PRINT "Auf Wiedersehen"
300 '
310 END
```

Listing 2: So sieht es schon besser aus

Listing 2 zeigt eine andere Lösung. Sie ist wesentlich übersichtlicher und auch noch von anderen Programmierern zu lesen.

Beide Programme stellen eine Art Kontoverwaltung dar. Der Benutzer wird nach der Summe gefragt, die er abheben will. Ist nicht mehr genug Geld auf dem Konto, so rückt die Bank nichts raus. Ansonsten zählt der Rechner den Kontostand um den betreffenden Betrag herunter. Sie werden feststellen, daß das erste Listing ziemlich unübersichtlich ist. Wäre auf diese Art ein größeres Programm geschrieben, würde man sich hoffnungslos darin verstricken.

Ordnung ist das...

Zu den einzelnen Formen der Strukturierung möchten wir nicht vergessen eine Möglichkeit aufzuzeigen, um Programmteile auf dem Papier darzustellen. Dies hat mehrere Gründe.

Bei der Programmierung sollte man generell versuchen sein Programm, sei es auch noch so lang, in einzelne, kleinere Unterprogramm zu unterteilen. Erstens kann man solche Unterprogramme mehrfach, von verschiedenen Stellen des Programms aufrufen und spart sich somit das mehrfache Schreiben gleicher Programmteile, zweitens wird ein Programm sofort übersichtlicher, wenn man es in Einzelprobleme aufteilt. Solche Unterprogramme kann man dann so universell gestalten, daß man sie auch in anderen Programmen verwenden kann, also der erste Schritt in Richtung modularer Programmierung.

Erst aufs Papier

Bei der Entwicklung komplizierterer Programme wird ein jeder feststellen, daß man dieses nicht einfach so schreiben kann. Dazu ist meist die Entwicklung auf dem Papier der eigentlichen Programmierung vorangestellt. Dazu gibt es mehrere Möglichkeiten. Für höhere Programmiersprachen ist das Nassi-Shneidermann Diagramm, kurz Struktogramm genannt, am geeignetsten, bei Assembler ein Flußdiagramm.

Das Struktogramm

Beginnen wir ganz einfach. Ein Programm wie das in Listing 3 besteht nur aus einer Reihe von fortlaufenden Befehlen, die der Reihe nach, also von oben nach unten abgearbeitet werden. In einem Struktogramm sieht das folgendermaßen aus (Bild 1). Jeder Kasten stellt einen Befehl, bzw. eine Anweisungsgruppe dar.

Wichtig ist, daß in solchen Diagrammen natürlich keine Basicsyntax stehen muß, sondern eher Klartext, z. B. 'Eingabe Wert' statt 'INPUT A'. Dies ist äußerst praktisch, da man dann ein

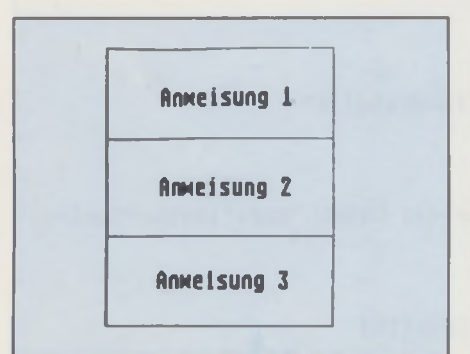


Bild 1: mehrere Anweisungen hintereinander

```
100 ' Beispiel ohne Schleifen
110 '
120 INPUT "Gib zwei Zahlen ein:  ";A,B
130 C=A+B
140 PRINT A;" +";B;" +";" =";C
150 '
160 END
```

Listing 3: Ein einfaches Programm ohne Schleifen

Struktogramm ohne Probleme in verschiedenen Hochsprachen umsetzen kann, also egal ob Pascal, Modula-2, Basic oder C.

Wiederhole bis Bedingung

Aber kommen wir zurück zu weiteren Formen des Struktogramms. Da wäre zunächst eine Schleife – genannt die REPEAT UNTIL-Schleife. Sie wiederholt einen Programmblock, also mehrere zusammenhängende Befehle, solange, bis eine Bedingung erfüllt ist.

```
REPEAT
   Anweisung 1
   Anweisung 2
   Anweisung 3
UNTIL Bedingung
```

Im Struktogramm ist dies wie in Bild 2 gekennzeichnet. Der Basic-Interpreter läuft zunächst in die Schleife hinein, führt die Anweisungen aus und trifft dann die Entscheidung, ob er wieder zum Schleifenanfang (vor Anweisung 1) verzweigen, oder die Schleife verlassen soll. Man spricht deshalb

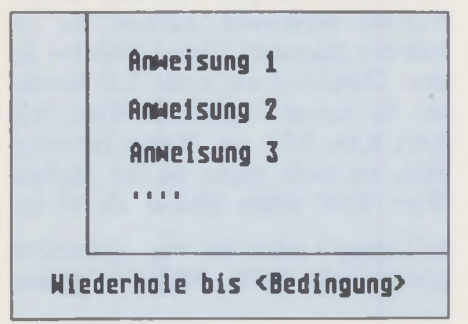


Bild 2: REPEAT...UNTIL, Wiederhole bis Ende

auch von einer nichtabweisenden Schleife. Im Beispiel hierzu (Listing 4) werden per Zufall Rechtecke mit wechselnden Füllmustern gezeichnet, und zwar solange, bis irgendeine Taste gedrückt wurde.

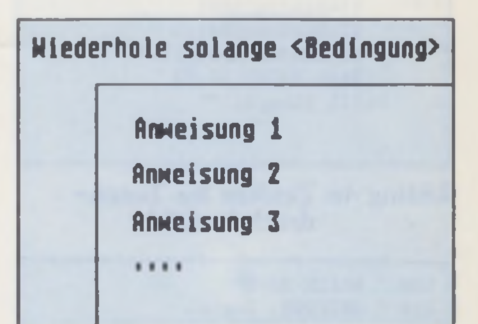


Bild 3: WHILE...WEND, Wenn Bedingung dann Wiederhole

Wenn Bedingung dann Wiederhole

Die nächste Schleife ist prinzipiell sehr ähnlich, nur prüft sie bereits am Anfang, ob die Bedingung erfüllt ist und läuft nur dann in den Programmblock innerhalb der Schleife hinein. Ist die Bedingung nicht erfüllt, so wird das Programm hinter dem Blockende fortgesetzt. Der Name 'WHILE WEND'. Das zugehörige Struktogramm sehen Sie in Bild 3.

```
WHILE Bedingung
   Anweisung 1
   Anweisung 2
   Anweisung 3
WEND
```

Auch wenn diese Schleifen ähnlich erschienen, so ist der Unterschied doch sehr wichtig (siehe Listing 5 und 5a).

In diesem Beispiel wird zunächst mittels der Fileselectorbox ein File angewählt. Dieses File, welches zweckmäßigerweise ein Textfile sein sollte, wird sodann geöffnet und Buchstabe für Buchstabe gelesen und gleich wieder auf den Bildschirm ausgegeben. Hat

```
100 ' Beispiel REPEAT UNTIL
110 ' OMIKRON. Basic
120 '
130 REPEAT
140   ' Hauptprogramm
150   FILL STYLE =2, RND(20)+1
160   PBOX RND(400), RND(200), RND(200)+1, RND(200)+1
170 UNTIL INKEY$ <>""
180
```

Listing 4: Zeichne bis Tastendruck in OMIKRON

```
' Beispiel REPEAT UNTIL
' GFA Basic
'
Repeat
  ' Hauptprogramm
  Deffill ,2,Random(20)+1
  X1=Random(400)
  Y1=Random(400)
  X2=Random(600)+1
  Y2=Random(400)+1
  Pbox X1,Y1,X2,Y2
Until Inkey$<>""
```

Listing 4a: Zeichne bis Tastendruck in GFA

```
100 ' WHILE WEND
110 ' OMIKRON. Basic
120 '
130 Pfad$="\*.*": MOUSEON
140 FILESELECT (Pfad$,Name$,Ok)
150 IF Ok THEN
160   OPEN "I",1,Name$
170   WHILE NOT EOF(1)
180     A$= INPUT$(1,1)
190     PRINT A$;
200     REM LPRINT A$;'Auf Drucker
210   WEND
220 ENDIF
230
```

Listing 5: Der Filegucker in OMIKRON

```
' WHILE WEND
' GFA Basic
'
Fileselect "*.*","",Name$
If Exist(Name$) Then
  Open "I",#1,Name$
  While Not Eof(#1)
    A$=Input$(1,#1)
    Print A$;
    Rem LPRINT A$;'Auf Drucker
  Wend
Endif
```

Listing 5a: Der Filegucker in GFA

man einen Drucker, so kann man das File auch sofort auf den Drucker geben. Aber zurück zur WHILE WEND Schleife. Sie ist in diesem Fall sehr wichtig, da Sie mit Hilfe der EOF (End Of File) Funktion erkennt, ob das File schon zu Ende ist oder man noch ein Zeichen lesen kann. Ist noch ein Zeichen zu lesen, so wird die Schleife durchlaufen und, wie oben beschrieben, ein von Diskette gelesenes Zeichen auf den Bildschirm gedruckt. Ist man bereits an das Fileende gekommen, so wird die Schleife umgangen. Wichtig ist dies, da ansonsten das Programm mit einer Fehlermeldung abbrechen würde. Wo nichts ist, kann man auch nichts lesen.

Von Bis

Eine spezielle Form der Schleife ist die FOR NEXT-Schleife. Sie sorgt dafür, daß ein Programmblock eine vorgegebene Anzahl mal durchlaufen wird. Sie ist die sogenannte Urbasic Schleife, die schon seit Geburt der Sprache Basic darin enthalten ist.

```
FOR Variable = Startwert TO Endwert
   Anweisung 1
   Anweisung 2
   Anweisung 3
NEXT Variable
```

Ein kleines Beispiel:

```
For I=1 To 10
  Print I
Next I
```

Eine Variable, in diesem Fall I, wird als Kontrollvariable benutzt. Sie bekommt einen Startwert und zählt diesen bis zum Endwert hoch und zwar standardmäßig um Eins. Benötigt man eine andere Schrittweite, so kann man diese optional verändern, z. B.:

For I=1 To 10 Step 3

oder

For Zaehler=10 To 9 Step −0.33

Im ersten Beispiel bekommt die Variable 'I' den Wert 1 und wird bei jedem Durchlauf um 3 erhöht. Sie nimmt also die Werte '1,4,7,10' an.

Im zweiten Beispiel wird eine andere Variable verwendet 'Zaehler'. Sie erhält den Startwert 10 und zählt bei jedem Durchlauf um exakt 0,33 herunter. Sie nimmt folglich die Werte '10, 9.67, 9.34, 9.01' an. Weiter herunter zählt sie nicht mehr, da der nächste Wert '8.68' schon kleiner als '9' ist.

In Listing 6 sehen Sie zwei ineinander geschachtelte FOR...NEXT-Schleifen. Erst wenn die Innere durchlaufen ist (hier von Spalte=1...15) springt der Interpreter wieder in die Äußere und erhöht die dortige Variable (Reihe). Auf diese Art wird eine einfache Multiplikationstabelle auf dem Bildschirm ausgegeben. Vielleicht versucht sich jemand einmal an einer Logarithmentabelle.

```
' Beispiel For..Next
' GFA. Basic
'
Cls
Print Tab(30);"Multiplikationstabelle"
'
For Reihe=1 To 15
  For Spalte=1 To 15
    Print At(5+Spalte*4,2+Reihe);
    Print Using "####",Reihe*Spalte
  Next Spalte
Next Reihe
```

Listing 6a: Das 1★1 in GFA

Abweisend oder nicht?

Jetzt ist es wieder einmal soweit. Unsere zwei Interpreter GFA- und OMIKRON. zeigen unterschiedliches Verhalten. Während OMIKRON.Basic eine abweisende FOR NEXT-Schleife enthält, zeigt sie sich bei GFA-Basic nichtabweisend. Im Klartext heißt das: in GFA-Basic wird eine FOR NEXT Schleife immer mindestens einmal durchlaufen, egal, ob die Endbedingung schon überschritten ist oder nicht. Darauf muß man im Einzelfall achten, da das Programm sonst leicht Fehler verursacht.

Ein Beispiel:

```
Start=7
Ende=5
For I=Start To Ende
  Print I
Next I
```

```
100 ' Beispiel For..Next
110 ' OMIKRON. Basic
120 '
130 CLS
140 PRINT TAB (30);"Multiplikationstabelle"
150 '
160 FOR Reihe=1 TO 15
170   FOR Spalte=1 TO 15
180     PRINT @(2+Reihe,5+Spalte*4); USING "####";Reihe*Spalte
190   NEXT Spalte
200 NEXT Reihe
210 HCOPY TEXT
220 LPRINT CHR$(10); CHR$(10); CHR$(10)
230 LLIST
```

Listing 6: Das 1★1 in OMIKRON

Multiplikationstabelle

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	4	6	8	10	12	14	16	18	20	22	24	26	28	30
3	6	9	12	15	18	21	24	27	30	33	36	39	42	45
4	8	12	16	20	24	28	32	36	40	44	48	52	56	60
5	10	15	20	25	30	35	40	45	50	55	60	65	70	75
6	12	18	24	30	36	42	48	54	60	66	72	78	84	90
7	14	21	28	35	42	49	56	63	70	77	84	91	98	105
8	16	24	32	40	48	56	64	72	80	88	96	104	112	120
9	18	27	36	45	54	63	72	81	90	99	108	117	126	135
10	20	30	40	50	60	70	80	90	100	110	120	130	140	150
11	22	33	44	55	66	77	88	99	110	121	132	143	154	165
12	24	36	48	60	72	84	96	108	120	132	144	156	168	180
13	26	39	52	65	78	91	104	117	130	143	156	169	182	195
14	28	42	56	70	84	98	112	126	140	154	168	182	196	210
15	30	45	60	75	90	105	120	135	150	165	180	195	210	225

Bild 4: So sieht die Tabelle aus

Obwohl der Endwert schon überschritten ist, läuft GFA-Basic in die Schleife hinein und führt die Befehle darin aus, ob die Bedingung erfüllt war oder nicht.

OMIKRON.Basic kümmert sich selbst darum und läuft nur wenn es soll.

Wenn dann ansonsten

Ein anderes Bestandteil der strukturierten Programmierung ist die 'IF THEN ELSE' Abfrage. Die direkte Übersetzung 'WENN DANN ANSONSTEN' sagt schon viel über den Zweck aus.

```
IF Bedingung Then
   Anweisung A1
   Anweisung  A2
     ..
ELSE
   Anweisung B1
   Anweisung B2
     ..
ENDIF
```

Im Struktogramm wird dies noch deutlicher (siehe Bild 5). Wenn die Bedingung erfüllt ist, verzweigt das Programm in Block A, ansonsten in Block B. Auch hier wird das Struktogramm immer von oben nach unten durchlaufen. Ein Sprung nach oben ist nicht möglich. Benötigt man den ELSE-Block nicht, so läßt man ihn einfach weg, und der Platz innerhalb Block B bleibt leer.

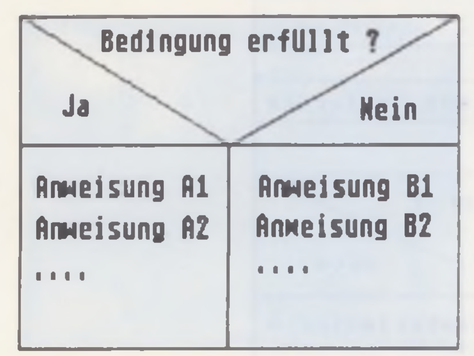


Bild 5: IF...THEN...ELSE, die Fallabfrage

Ein weiteres Beispiel (Listing 7) zeigt zwei IF-Abfragen, die wiederum von einer REPEAT...UNTIL-Schleife umgeben sind. Hört sich kompliziert an, doch das Listing überzeugt vom Gegenteil. Es handelt sich hierbei um ein einfaches Malprogramm, welches bei gedrückter linker Maustaste auf dem Bildschirm zeichnet und bei gedrückter linker Maustaste auf dem Bildschirm löscht.

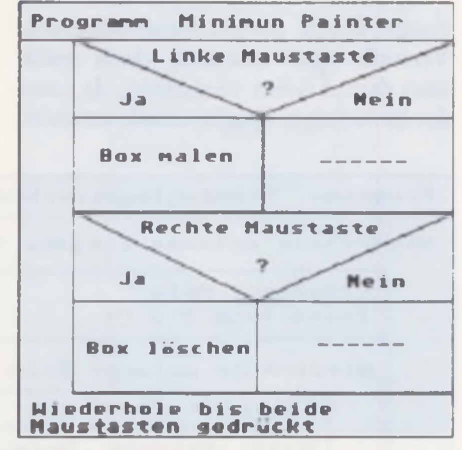


Bild 6: Der Minimum Painter im Struktogramm

Wie man in Listing 1+2 sehen kann, tut das Einrücken innerhalb der Schleifen der Lesbarkeit der Programme unheimlich gut. GFA-Basic rückt ja bekanntlich automatisch ein. Der Anfänger wird dies zu schätzen wissen, da er Fehler leichter findet und alles gleich ordentlich aussieht. Allerdings hat dies zur Folge, daß nur noch ein Befehl in jeder Zeile stehen darf. Das führt dazu, daß jede Befehlsfolge, egal ob sinnvoll oder nicht in viele Zeilen aufgeteilt werden muß. Das Programm geht somit in die Länge und der Programmiererfreiheit ist Einhalt geboten. Aber jedem das Seine.

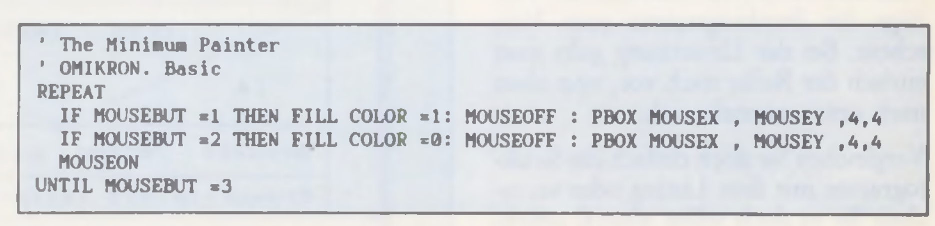
```
' The Minimum Painter
' GFA Basic
Repeat
  If Mousek=1 Then
    Deffill 1
    Pbox Mousex,Mousey,Mousex+4,Mousey+4
  Endif
  If Mousek=2 Then
    Deffill 0
    Pbox Mousex,Mousey,Mousex+4,Mousey+4
  Endif
Until Mousek=3
```

Listing 7a: Der Minimum Painter in GFA

Endlos schleifen

GFA-Basic bietet noch eine weitere Schleifenvariante, die spezielle Endlosschleife.

```
DO
   Anweisung 1
   Anweisung 1
      ..
      ..
   Anweisung n
LOOP
```



```
' The Minimum Painter
' OMIKRON. Basic
REPEAT
  IF MOUSEBUT =1 THEN FILL COLOR =1: MOUSEOFF : PBOX MOUSEX , MOUSEY ,4,4
  IF MOUSEBUT =2 THEN FILL COLOR =0: MOUSEOFF : PBOX MOUSEX , MOUSEY ,4,4
  MOUSEON
UNTIL MOUSEBUT =3
```

Listing 7: Der Minimum Painter in OMIKRON

Der Block (Anweisung 1 bis Anweisung n) wird unendlich oft wiederholt. In OMIKRON Basic existiert sie nicht, läßt sich das leicht mit einer der anderen Schleifen imitieren.

```
REPEAT
  Anweisung 1
  Anweisung 2
  ..
  ..
  Anweisung n
Until 0
```

oder etwas verständlicher ausgedrückt

```
REPEAT
..
..
UNTIL immer_und_ewig
```

'immer_und_ewig' ist in diesem Fall eine Variable, die, da sie wohl kaum benutzt wird, den Wert Null enthält.

Alles zusammen

Nun wollen wir einmal ein komplexeres Programm schreiben, welches mehrere der eben besprochenen Schleifentypen beinhaltet (siehe Listing 8).

Das Programm zerlegt eine Zahl in seine Primfaktoren, also in Zahlen, die sich durch keine anderen mehr teilen lassen. Der Aufbau im Struktogramm ist in Bild 8 zu sehen. Auch wenn es etwas länger ist, so erkennt man doch den Zusammenhang zwischen dem Programm im Rechner und auf dem Papier.

```
 ***  Primfaktorzerlegung  ***
 Bitte geben Sie eine Zahl ein
? 653426

 653426 =  2 *  239 *  1367

Weiter [J/N]
```

Bild 7: So wird eine Zahl in Primfaktoren zerlegt

Man sieht: Erst jetzt kommen die Vorzüge des Struktogramms zum Vorschein. Bei der Umsetzung geht man einfach der Reihe nach vor, von oben nach unten versteht sich.

Vergleichen Sie doch einfach das Struktogramm mit dem Listing oder versuchen Sie es doch selbst einmal umzusetzen.

```
' Beispiel Primfaktorzerlegung
' OMIKRON. und GFA Basic
'   keine Unterschiede
'
While Upper$(Antwort$)<>"N"
  Cls
  Print "  ***  Primfaktorzerlegung  ***"
  Print "  Bitte geben Sie eine Zahl ein "
  Input Prim
  Print
  Print Prim;" = ";
  While Prim>1
    Teiler=2
    While Prim Mod Teiler<>0
      Teiler=Teiler+1
    Wend
    '
    If Prim=Teiler Then
      Print Teiler
    Else
      Print Teiler;" * ";
    Endif
    '
    Prim=Prim\Teiler
  Wend
  Print
  Print "Weiter [J/N]"
  Antwort$=Input$(1)
Wend
```

Listing 8: Primzahlenzerlegung ohne Kopfzerbrechen

Kombination

Wie schon in den letzten Listings zu sehen war, werden natürlich immer viele dieser Strukturelemente in einem Programm benötigt. Zwar kann man einige durch programmiertechnische Verrenkungen ersetzen, doch sollte man darauf lieber verzichten, da sonst der berüchtigte Spaghetticode entsteht.

Als Fazit wäre anzumerken, daß mit solchen Fähigkeiten das Programmieren in Basic wieder Spaß macht. Man sollte versuchen, alle diese angebotenen Möglichkeiten zu nutzen, denn diese bekommt man ansonsten nur in Sprachen wie PASCAL, C oder MODULA 2 geboten.

HS + MM

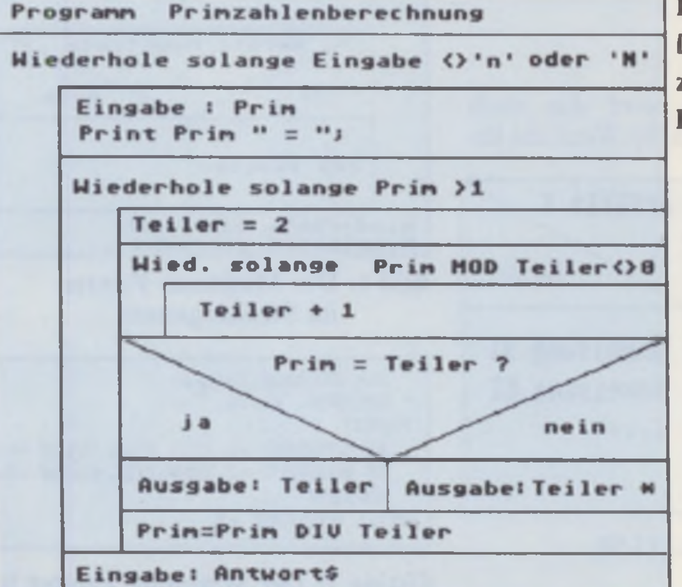


Bild 8: Das Struktogramm zur Primfaktorzerlegung

Einkaufsführer

Hier finden Sie Ihren Atari Fachhändler

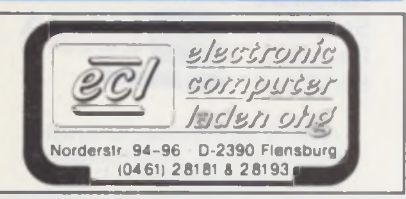
Anzeigenschluß Heft 1/88: 13. Nov. 1987

2350 Neumünster



Klosterstraße 2 · 2350 Neumünster
Telefon (0 43 21) 4 39 33

2390 Flensburg

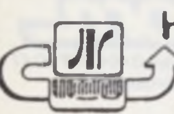


2800 Bremen

PS-DATA
Doventorsteinweg 41
2800 Bremen
Telefon 04 21 - 17 05 77

2850 Bremerhaven

HEIM- UND PERSONALCOMPUTER



Kurt Neumann
Georgstraße 71
2850 Bremerhaven
Tel. 04 71 / 30 21 29

HARDWARE · SOFTWARE · PAPIERWARE

2940 Wilhelmshaven

Radio Tiemann
ATARI-Systemfachhändler
Markstr. 52
2940 Wilhelmshaven
Telefon 0 44 21 - 2 61 45

2950 Leer



- HARDWARE-SOFTWARE
- SYSTEM-ENTWICKLUNG
- ORGANISATION
- EDV-SCHULUNG
- EDV-BERATUNG
- SERVICE-WARTUNG

Augustenstraße 3 · 2950 Leer
Telefon 04 91 - 45 89

3000 Hannover

COM DATA
Am Schiffgraben 19 · 3000 Hannover 1
Telefon 05 11 - 32 67 36

3000 Hannover



DATALOGIC COMPUTERSYSTEME
ATARI ST- COMPUTER · BERATUNG
HARDWARE · SERVICE
SOFTWARE · VERKAUF
CALENBERGER STR. 26
3000 HANNOVER 1
TEL. 0511 · 32 64 89

trendDATA Computer
IBM · EPSON · TRIUMPH ADLER
HEWLETT PACKARD · ATARI etc.
trendDATA Computer GmbH
Am Marstall 18–22 · 3000 Hannover 1
Telefon (05 11) 1 66 05-0

3040 Soltau

F & T Computervertrieb
Am Hornberg 1
(Industriegeb. Almhöhe)
3040 Soltau
Tel. 0 51 91 / 1 65 22

3100 Celle

Ludwig Haupt jr.
Büro-Einkaufs-Zentrum
Gerhard-Kamm-Straße 2
Ruf 8 30 45, Postfach 140
3100 Celle

3150 Peine

Wieckenberg & Schrage GmbH
Computertechnik
Hard- u. Software
Woltorfer Str. 8, 3150 Peine
Tel. 0 51 71 / 60 52/3 o. 0 51 73 / 79 09

3170 Gifhorn

COMPUTER-HAUS
GIFHORN
Braunschweigerstr. 50
3170 Gifhorn
Telefon 0 53 71 - 5 44 98

3300 Braunschweig

COMPUTER STUDIO
BRAUNSCHWEIG
Rebenring 49-50
3300 Braunschweig
Tel. (05 31) 33 32 77 / 78

3400 Göttingen



3400 Göttingen-Weende
Wagenstieg 14 – Tel. 05 51 / 38 57-0

3470 Höxter



3500 Kassel

Hermann Fischer GmbH
autorisierter ATARI-Fachhändler
Rudolf-Schwander-Str. 5-13
3500 Kassel
Tel. (05 61) 70 00 00

RUDOLPH
Computersysteme
& Bürotechnik
Einzel+Großhandel Im+Export
Telefon: 0561/472737
Telex Fax Btx 0561472739
Computerstudio: 3500 Kassel
Frankfurterstraße 311
Versand: 3501 Hoof Pf. 1160

3550 Marburg

L W M
COMPUTER SERVICE
Bahnhofstraße 26 b
3550 Marburg/Lahn
✆ 0 64 21 - 6 22 36

4000 Düsseldorf

BERNSHAUS GmbH
Bürotechnik – Bürobedarf
Cäcilienstraße 2
4000 Düsseldorf 13 (Benrath)
Telefon 02 11 - 71 91 81

HOCO
EDV ANLAGEN GMBH
Flügelstr. 47
4000 Düsseldorf
Tel. 02 11 - 77 62 70

4050 Mönchengladbach



Hindenburgstr. 249
4050 Mönchengladbach
Tel. 0 21 61 - 187 64

4150 Krefeld



- Service-Center
- Hardware
- Software
- Erweiterungen

COP
Computer Service GmbH
Lewerentz Str. 111
4150 Krefeld
Tel. 0 21 51 / 77 30 42

4200 Oberhausen

LaSch
das Buch und Software Haus
Inh. Rainer Langner u. Franz Schnitzler GbR
Nohlstraße 76 · 4200 Oberhausen 1
Telefon 02 08 / 80 90 14

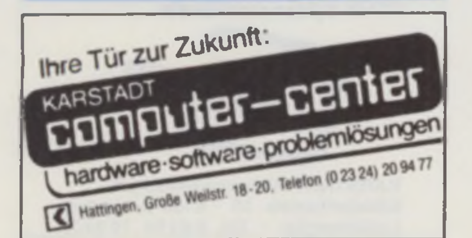
4300 Essen

ATARI Systemfachhändler

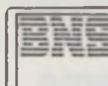


KARSTADT Aktiengesellschaft
Limbecker Platz 4300 Essen 1
Tel.: (02 01) 17 63 99

4320 Hattingen



4330 Mülheim



Computer und Bürotechnik
Vertriebsgesellschaft mbH
Dickswall 78 4330 Mülheim Telefon 02 08 / 3 40 34

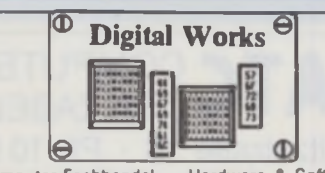
Tandon Computer
Computer Hard- und Software auch im Leasing
Computerkurse für Anfänger und Fortgeschrittene
VICTOR
SEL-Fernkopierer
NEC ATARI OKI

4400 Münster



COMPUTER SYSTEME GMBH
Daimlerweg 39 - 4400 Münster
Telefon 02 51 / 71 99 75 - 9

4410 Warendorf



Computer-Fachhandel – Hardware & Software
Jörg Kellert – Helmut Müller GbR
Brünebrede 17 · 4410 Warendorf
Tel. 0 25 81 / 6 11 26

4422 Ahaus

ATARI · Epson · Fujitsu · Molecular · NCR · Tandon · Schneider · Star
OCB

OCB-Computershop	OCB-Hard- und Software
Wallstraße 3	Wessumerstraße 49
4422 Ahaus	4422 Ahaus
Tel. 0 25 61 / 50 21	Tel. 0 25 61 / 50 21

4430 Steinfurt

ATARI SCHNEIDER STAR NEC SEIKOSHA PANASONIC EPSON
Computer
Büromaschinen
Service
Telefon 02551/2555
Tecklenburger Str. 27 · 4430 Steinfurt

4500 Osnabrück

Heinicke-Electronic
Kommenderiestr. 120 · 4500 Osnabrück
Telefon 05 41 - 8 27 99
Wir liefern Micro-Computer seit 1978

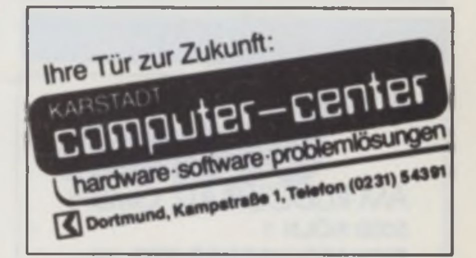
4600 Dortmund

Bürostudio BOLZ
Brauhausstraße 4 · 4600 Dortmund
Telefon 02 31 - 52 77 13-16

4600 Dortmund



Atari, Basis, Schneider, Tandy, Brother, Star, Memorex, BASF, Verbatim
cc Computer Studio GmbH
Software-Hardware-Beratung
Service-Eilversand
Ihre Ansprechpartner:
v. Schablinski
Jan P. Schneider
Elisabethstraße 5
4600 Dortmund 1
T. 0231/528184 · Tx 822631 cccsd



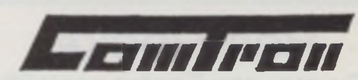
Elektronik
Computer
Fachliteratur
ATARI-System-Fachhändler
4600 Dortmund 1, Güntherstraße 75, Tel. (02 31) 57 22 84



4620 Castrop-Rauxel

R. Schuster Electronic
OBERE MÜNSTERSTR. 33-35 · (0 23 05) 1770 · 4620 CASTROP RAUXEL
ATARI
System-Fachhändler

4650 Gelsenkirchen-Horst



Hard- und Software, Literatur
Bauteile, Service, Versand
Groß- und Einzelhandel
Poststr. 15 · 4650 Gelsenkirchen-Horst
Tel. 0209/52572

4700 Hamm



4800 Bielefeld

hardware software organisation service
CSF
CSF COMPUTER & SOFTWARE GMBH
Heeper Straße 106 – 108
4800 Bielefeld 1
Tel. (05 21) 6 16 63

4800 Bielefeld

Carl-Severing-Str. 190
4800 Bielefeld 14

Telefon: 05 21 / 45 99-150
Telex : 9 37 340 krab d
Telefax: 05 21 / 45 99-123

MICROTEC
Software
Hardware
Beratung
Service

5000 Köln

BÜRO MASCHINEN
braun
AM RUDOLFPLATZ GmbH
5000 KÖLN 1
RICHARD-WAGNER-STR. 39
RUF: 02 21 / 21 91 71

5010 Bergheim



EDV-Beratung · Organisation
Programmierung · Home/Personal-Computer
Software · Zubehör · Fachliteratur
Zeppelinstr. 7 · 5010 Bergheim
Telefon 0 22 71 - 6 20 96

5060 Bergisch-Gladbach

Computer Center
Buchholzstraße 1
5060 Bergisch-Gladbach
Telefon 0 22 02 - 3 50 53

5090 Leverkusen

Rolf Rocke
Computer-Fachgeschäft
Auestraße 1
5090 Leverkusen 3
Telefon 0 21 71 / 26 24

5200 Siegburg

Computer Center
Luisenstraße 26
5200 Siegburg
Telefon 0 22 41/6 68 54

5400 Koblenz

SCHMITT
COMPUTERSYSTEME
Casinostraße 40
5400 Koblenz
✆ 02 61 - 3 65 28

5457 Straßenhaus

DR. AUMANN GMBH
Computer-Systeme
Schulstr. 12
5457 Straßenhaus
Telefon 0 26 34 - 40 81/2

5500 Trier



Güterstr. 82 - 5500 Trier
☎ 06 51 - 2 50 44
Fordern Sie unsere Zubehör-Liste an.

5540 Prüm

ATC COMPUTER
J. M. ZABELL
Ritzstraße 13 · Pf. 10 51
5540 PRÜM
– Tel.: 0 65 51 - 30 39 –

5600 Wuppertal

Jung am Wall
Wall 31—33
5600 Wuppertal 1
Telefon 02 02/45 03 30

MEGABYTE
Computer Vertriebs GmbH
Friedrich-Engels-Allee 162
5600 Wuppertal 2 (Barmen)
Telefon (02 02) 8 19 17

5630 Remscheid

COM SOFT
Schelderstr. 12 · 5630 Remscheid
Telefon (0 21 91) 2 10 33 - 34

5800 Hagen



Vertragshändler Axel Böckem
Computer + Textsysteme
Eilper Str. 60 (Eilpezentrum) · 5800 Hagen
Tel. 0 23 31 / 7 34 90

5900 Siegen



Siegen · Weidenauer Str. 72 · ☎ 02 71/7 34 95

6000 Frankfurt

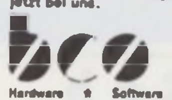
Müller & Nemecek
Kaiserstraße 44
6000 Frankfurt/M.
Tel. 0 69 - 23 25 44

GES-COMPUTER
GESELLSCHAFT FÜR EDV UND SOFTWARE mbH

Filiale Frankfurt	Filiale Hanau
Hartmann-Ibach-Str. 63	Steinheimer Str. 22
6000 Frankfurt 60	6450 Hanau
Tel.: (0 69) 46 20 41	Tel.: (0 61 81) 2 48 26

WAIZENEGGER
Büroeinrichtungen
Kaiserstraße 41
6000 Frankfurt/M.
☎ 0 69 / 2 73 06 - 0

ATARI
... wir machen Spitzentechnologie preiswert.
Vertragshändler
Jetzt bei uns.



Büro-Computer + Organisations GmbH
Oederweg 7–9
6000 Frankfurt/M. 1
☎ (0 69) 55 04 56-57
Hardware * Software * Beratung * Service

SCHMITT
COMPUTERSYSTEME
Hanauer Landstr. 417
6000 Frankfurt/M. 1
Tel: 0 69/41 02 71
FAX: 0 69/42 57 71

6100 Darmstadt

Heim
Büro- und Computermarkt
Heidelberger Landstraße 194
6100 Darmstadt-Eberstadt
✆ 0 61 51 / 5 60 57

ATARI Systemfachhändler



KARSTADT Aktiengesellschaft
Elisabethenstr. 15 · 6100 Darmstadt
Luisencenter · Tel. 0 61 51 - 10 94 20

6100 Darmstadt

SCHMITT COMPUTERSYSTEME
Mühlstraße 76
6100 Darmstadt
✆ 06151-245574

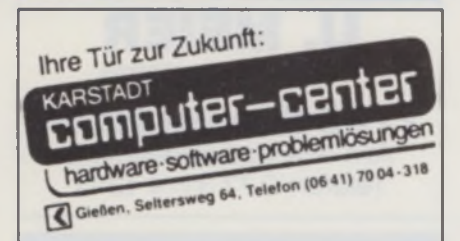
6200 Wiesbaden

SCHMITT COMPUTERSYSTEME
Rheinstraße 41
6200 Wiesbaden
✆ 06121-307330

6240 Königstein

KFC COMPUTERSYSTEME
Wiesenstraße 18
6240 Königstein
Tel. 06174-3033
Mail-Box 06174-5355

6300 Gießen

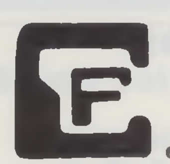


Schneider ATARI Commodore
BAUMS
BÜRO · ORGANISATION
Bahnhofstr. 26 · 6300 Gießen
Telefon: 0641/71096



Thomas Heß
Asterweg 10 · 6300 Gießen
Telefon 0641/39153

6330 Wetzlar



Fachmarkt für Computer u. Unterhaltungs-electronic in Wetzlar.
Einkaufszentrum Bahnhofstraße, Tel. (06441) 48566

6400 Fulda

Schneider ATARI Commodore
WEINRICH
BÜRO · ORGANISATION
Ronsbachstraße 32 · 6400 Fulda
Telefon: 0661/492-0

6457 Maintal

Landolt-Computer
Beratung · Service · Verkauf · Leasing
Wingertstr. 112
6457 Maintal/Dörnigheim
Telefon 06181-45293

6500 Mainz

ELPHOTEC
Computer Systeme
Ihr Atari Systemhändler
mit eigenem Service-Center
Walpodenstraße 10
6500 Mainz
Telefon 06131-231947

SCHMITT COMPUTERSYSTEME
Karmeliterplatz 4
6500 Mainz
✆ 06131-234223

6520 Worms

ORION
Computersysteme GmbH
Friedrichstraße 22
6520 WORMS
Tel. 06241/6757-6758

6700 Ludwigshafen

MKV Computermarkt
Bismarck-Zentrum
6700 Ludwigshafen
Telefon 0621-525596

6720 Speyer

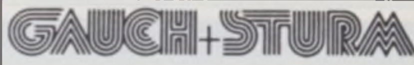
MKV Computermarkt
Gilgenstraße 4
6720 Speyer
Telefon 06232-77216

6750 Kaiserslautern

C.O.S
Computer
Organisation GmbH
6750 Kaiserslautern
Telefon (0631) 92051



6800 Mannheim



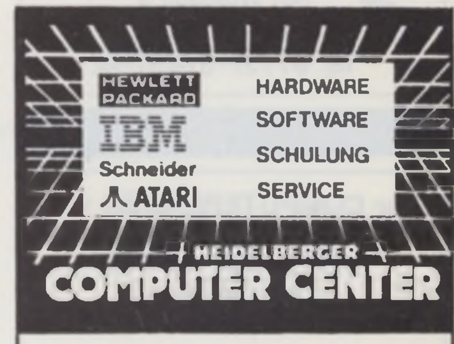
Computersysteme + Textsysteme
6800 Mannheim 24
Casterfeldstraße 74-76
☎ (0621) 850040 · Teletex 6211912



L 14, 16-17
6800 Mannheim 1
Tel. (0621) 20983/84

6900 Heidelberg

JACOM COMPUTERWELT
Hardware · Software
Schulung · Service
Mönchhofstraße 3 · 6900 Heidelberg
Telefon 06221/410514-550

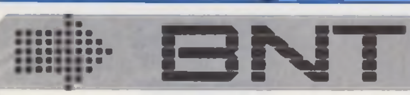


Bahnhofstr. 1 / Tel. 06221 / 27132

7000 Stuttgart

SCHMITT COMPUTERSYSTEME
Tübingerstr. 18
7000 Stuttgart

7000 Stuttgart



COMPUTERFACHHANDEL

Ihr starker Partner in Stuttgart

BNT Computerfachhandel GmbH Marktstr. 48 1. Stock 7000 Stuttgart-Bad Cannstatt in der Fussgängerzone direkt beim Rathaus Telefon (0711) 558383

7022 L-Echterdingen

Autorisierter ATARI-System-Fachhändler

ATARI ST



Matrai Computer GmbH
Bernhauser Str 8
7022 L-Echterdingen
☎ (0711) 797049

7030 Böblingen

Verkauf – Service – Software



Norbert Hlawinka
Sindelfinger Allee 1
7030 Böblingen
Tel. 07031/226015

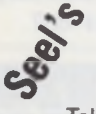
COMPUTER CENTER

7100 Heilbronn

Unser Wissen ist Ihr Vorteil

Walliser & Co.
Mönchseestraße 99
7100 Heilbronn
Telefon 07131/60048

Computer-Welt



Am Wollhaus 6
7100 Heilbronn
Tel. 07131-68401-02

7101 Flein

der **COMPUTERLADEN** von



In der Falterstraße
7101 Flein

Beratung, Service, Verkauf,
Software-Entwicklung
direkt beim Peripherie-Hersteller

ST-Einkaufsführer



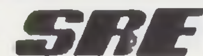
aktuell
preiswert
werbewirksam



7150 Backnang



7410 Reutlingen



Werner Brock
autorisierter Fachhändler f. ATARI, Schneider, Oki

Federnseestr. 17 · 7410 Reutlingen
Telefon: 07121/34287

7450 Hechingen

SRE

Gesellschaft für Datenverarbeitung mbh

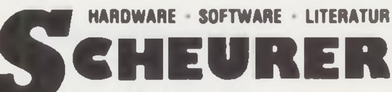
Computer · Drucker
Zubehör · Fachliteratur

Schloßplatz 3 · 7450 Hechingen
Telefon 07471/14507

7475 Meßstetten

Ihr ATARI-Systemhändler im Zollern-Alb-Kreis

HEIM + PC-COMPUTERMARKT



ATARI COMMODORE CUMANA DATA BECKER MULTITECH RITEMAN SCHNEIDER THOMSON

7475 Meßstetten 1 · Hauptstraße 10 · 07431/61280

7480 Sigmaringen



7500 Karlsruhe

papierhaus erhardt

Am Ludwigsplatz · 7500 Karlsruhe
Tel. 0721-1608-16

MKV GMBH

Rüppurer Str. 2d
7500 Karlsruhe
Telefon 0721-84613

7530 Pforzheim

DM Computer GmbH
Hard- & Software
Durlacherstr. 39
Tel. 07231-13939
7530 Pforzheim
Telex 783248

7600 Offenburg

FRANK LEONHARDT ELECTRONIC
Ihr Fachgeschäft für Microcomputer · Hifi · Funk

In der Jeuch 3
7600 Offenburg
Telefon 0781/57974

7640 Kehl/Rhein



Badstrasse 12
Tel. 07807/822
Telex: 752913
7607 NEURIED 2

Filiale:
Hauptstrasse 44
Tel. 07851/1822
7640 KEHL/RHEIN

ELEKTRO-MÜNTZER GmbH

7700 Singen

U. MEIER
Computersysteme

7700 Singen-Htwl.
Am Posthalterswäldle 8
Telefon 07731-44211

7730 VS-Schwenningen

BUS BRAUCH & SAUTER COMPUTER TECHNIK

Villinger Straße 85
7730 VS-Schwenningen
Telefon 07720/38071-72

7750 Konstanz

ATARI ★ PC's ★ SCHNEIDER

computer-fachgeschäft
rösler
Rheingutstr. 1 · ☎ 07531-21832

7800 Freiburg

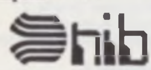
CDS
EDV-Service GmbH

Windausstraße 2
7800 Freiburg
Tel. 0761-81047

8500 Nürnberg

ATARI

Der ATARI - Systemfachhändler in Nürnberg
Softwareberatung und Entwicklung
Computer • Drucker • Zubehör • Fachliteratur



HIB Computer GmbH
Äußere Bayreuther Str. 72
8500 Nürnberg
Tel.:0911/515939

8520 Erlangen



Computerservice Decker
Meisenweg 29 - 8520 Erlangen
Telefon 0 91 31 / 4 20 76



8520 Erlangen
Dresdener Str. 5
Friedrichstr. 9
Tel.: 0 91 31 / 1 20 10

Zimmermann elektroland

8520 Erlangen
Nürnberger Straße 88
Tel. (0 91 31) 3 45 68

8500 Nürnberg
Hauptmarkt 17
Tel. (09 11) 2 07 98

8600 Bamberg



8700 Würzburg

HALLER GMBH
Fachgeschäft für
Mikrocomputer
Büttnerstraße 29
8700 Würzburg
Tel. 09 31 / 1 67 05

8700 Würzburg

SCHÖLL BÜROTEAM

Hardware · Software
Service · Schulung

computer center
am Dominikanerplatz
Ruf (09 31) 5 04 88

Müller-Anwendersoftware
Versandhandel für PC-Software
Textverarb., Graphik, Buchf.,
Spiele PD-Software erklärt u. a.
Matterstockstr. 23 a · 8700 Würzburg
Tel. 09 31 / 28 23 26

8720 Schweinfurt

Uhlenhuth GmbH
Computer + Unterhaltungselektronik
Albrecht-Dürer-Platz 2
8720 Schweinfurt
Telefon 0 97 21 / 6 52 1 54

8900 Augsburg

Adolf & Schmoll Computer
Schwalbenstr. 1 · 8900 Augsburg
Tel. (08 21) 52 85 33 oder 52 80 87
Wir sind außerdem autorisierte
Service-Fachwerkstatt für:
Schneider
commodore
ATARI

ATARI Systemfachhändler



KARSTADT Aktiengesellschaft
Bürgermeister-Fischer-Str. 6-10
8900 Augsburg · Tel. (08 21) 31 53-416

SCHMITT COMPUTERSYSTEME

Frauentorstr. 22
8900 Augsburg
✆ 08 21 - 15 42 68

8940 Memmingen

EDV-Organisation
Hard- + Software
Manfred Schweizer KG
Ulmer Str. 2, Tel. 0 83 31 / 1 22 20
8940 Memmingen

Österreich

A-1020 Wien

W + H.
Computerhandel Ges m.b.H.

Förstergasse 6/3/2 · 1020 Wien
Tel. 02 22 - 35 09 68

A-1040 Wien

Ihr ST-Fachhändler in Wien
Computer-Studio
Wehsner Gesellschaft m.b.H.

1040 Wien - Paniglgasse 18-20
Telefon 02 22 - 65 78 08, 65 88 93

A-8010 Graz



Schweiz

CH-1700 Fribourg

Softy Hard's Computershop
FRIDAT SA
Die ATARI ST Spezialisten

Grand Rue 42
CH-1700 Fribourg
Tel. 0041(0) 37 22 26 28

CH-2503 Biel

UE
URWA Electronic

Ihr ATARI ST Spezialist in
der Schweiz.
0 32 / 25 45 53

Lindenweg 24, 2503 Biel



Wo ist der ATARI Fachhändler in meiner Umgebung?

Der Einkaufsführer in ST-Computer

Anzeigenschluß:
Heft 12/87: 27.11.87
Heft 1/88: 18.12.87

Pouvez vous Français?

Ein Sprachlehrprogramm

Für all diejenigen, die etwas anderes als „C", „Pascal", oder „Logo" unter Sprachen verstehen, gibt es Programme, die es Ihnen ermöglichen Fremdsprachen besser zu beherrschen. „Pouvez vous francais" von Data Beutner KG gehört dazu, und besteht aus zwei (1 und 2) getrennt zu erhaltenen Teilen.

Laut Verpackungstext befasst sich das Programm ausschließlich mit der Verbesserung von typisch deutschen Fehlern, die aus den Gewohnheiten der mütterlichen Sprache resultieren. Als Beispiel: „pouvez-vous Francais" ist die direkte Übersetzung von „können Sie Französisch" – sie ist aber keineswegs richtig. Nach diesem Vorwort sucht man vergebens in der französischen Sprache.

Für welche Anwendergruppe ist das Programm nun geschrieben? Für den Anfänger oder für den ausgereiften Sprachstudenten? Ein Blick auf den Bildschirmausdruck klärt dies. Anfänger sind damit hoffnungslos überlastet, die fortgeschrittenen Französischbüffler hingegen finden in diesem Programm, das übrigens auch als Taschenbuch erschienen ist, eine gute Hilfe zum Verlernen typisch deutscher Fehler.

Das Äußere

Aus der dicken Verpackung blickt eine Diskette und eine dünne Anleitung, die den Schwierigkeiten des Programms angepasst ist, hervor. Die Anleitung von Teil 1 und 2 sind identisch, mit Ausnahme der Nummern. Teil 1 enthält zusätzlich ein beide Teile umfassendes Taschenbuch, welches schon seit geraumer Zeit im Handel ist. Hier zeigt sich, daß der Inhalt des Programms ausnahmslos diesem, übrigens sehr erfolgreichen Buch, entnommen ist.

Wenn man sieht, daß beide Programme identisch sind (bis auf die steigende Schwierigkeit der Übungen), fragt man sich, warum beide Disketten nicht in einem Pakett auf dem Markt gekommen sind.

Das Innere

Nach dem Laden bekommt man ein Bild das aus einem „Arbeitsfeld" und einem „Hilfefeld" besteht. Man wählt eine von 37 Übungen (je 15 Fragen, das ergibt 455 Fragen) aus, und schon geht es los: Das Programm schlägt einen Satz vor, dessen Richtigkeit in Frage gestellt wird. Eine positive Antwort bringt den Vergleich mit der Lösung. Bei einer negativen Antwort muß man dem Programm jedoch den korrigierten Satz anbieten, welches ihn prüft und gegebenenfalls die Korrektur korrigiert.

Man kann entweder einfach die fehlerhaften Wörter neuschreiben, oder gleich den ganzen Satz, mit allen Sonderzeichen der französischen Sprache, die dank geänderter Tastaturbelegung zugänglich sind, eingeben. Diese Eigenschaft ist einer der Pluspunkte dieses Programms.

Das Programm überprüft den Satz und bewertet gnadenlos alle Abweichungen von seinem „Standard" als Fehler. Man wird dadurch natürlich gefördert Französisch zu schreiben und muß dabei jede Kleinigkeit beachten.

Das „Hilfefeld" liefert erstmals das Urteil des Programms. Wenn gewünscht kann man nach Beantwortung der Frage Erläuterungen des Falles bekommen (Eine kurzgefasste Regel sowie ein paar Beispiele). Die anderen Hilfsfelder erläutern die Schreibweise, falls dies noch nötig ist.

Am Ende jeder Übung wird dann die Zahl der richtig-beantworteten Fragen gezeigt, und auf Wunsch ein Fehlerprotokoll ausgedruckt, welches allerdings nur die Nummer der falschen Antwort und nicht die Antwort selbst dem Drucker zuführt.

Mit steigender Nummer werden auch die Sätze schwieriger. Der steigende Schwierigkeitsgrad fordert recht gute Kenntnisse der französischen Sprache. Damit müssen die Anfänger gewarnt werden.

Die Bedienung

Durch seine Einfachheit ist dieses Programm sehr leicht zu handhaben. Diese Einfachheit ist aber gleichzeitig seine große Schwäche. Es fehlt jede Flexibilität. Es gibt leider, obwohl die französische Sprache recht vielfältig ist, nur eine richtige Antwort, die des Programms. Es ist ebenfalls nicht möglich, z. B. für einen Lehrer, selbst konzipierte Aufgaben zu gestalten. Es fehlt auch jede Möglichkeit seine Ergebnisse zu speichern um seinen Lernfortschritten zu folgen.

Kurz gesagt eine gute Idee, die durch die Möglichkeit die Texte zu schreiben einen Vorteil gegenüber dem Ursprungsbuch („Pouvez vous Francais" von Jacques Soussan erschienen im Rohwolt Verlag) hat. Leider haben die Programmierer es versäumt noch etwas an dem Rahmenprogrammm zu feilen, was dem inhaltlich guten Produkt gerecht wäre.

Der Lehrinhalt des Produktes ist empfehlenswert, und manch einer lernt am Bildschirm lieber als mit einem Buch.

Abgerundet wird dieses Produkt mit zwei weiteren Teilen zum Lernen der französischen Grammatik 'Maitre de Grammaire 1+2'. Das Rahmenprogramm ist identisch.

Der Preis eines Teiles liegt bei 89,– DM. Aufgrund der zu einfachen Programmgestaltung, obwohl bei einem



Bild 1: Die falsche Antwort und die richtige Lösung

solchen Produkt der Inhalt wichtiger ist als das Rahmenprogramm, ist er doch etwas hoch geraten. Der didaktisch gut aufgebaute Inhalt bietet dazu den Gegenpol.

Philippe Lerailler

Bezugsadresse
Data Beutner KG
Postfach 11 11
2217 Kellinghusen
Tel: 0 48 22 / 27 52

Weitere Produkte dieser Serie:
Speak you English 1+2
Englische Grammatik 1+2
Computerslang
Verkäuferenglisch
Handelsenglisch
Engl. f. Bank und Börse
Mathematik

Deus ex machina:



Es gibt in der Welt der Personalcomputer nur wenige Programme, von denen jederman sagt, man müsse sie gesehen haben. Den Flightsimulator von Sublogic zum Beispiel. Oder eben Steve, den ST EVent Editor für den Atari ST. In aller Kürze: das ist eine Textverarbeitung so schnell wie Tempus, ein Dateiprogramm so komfortabel wie Datamat, ein Graphiktool so luxeriös wie Degas und ein Spelling-Checker, wie man ihn sich schon immer wünschte. All dies, zu einem Paket zusammengeschnürt, ist Steve.

Beginnen wir mit einigen *Benchmarks* als Anregung. Wir haben einen Mustertext mit 40.000 Wörtern im Umfang von rd. 500 KByte erzeugt. Das Springen an den Anfang oder das Ende des Textes ist eine Sache von Sekundenbruchteilen. Mit enormer Geschwindigkeit verlaufen die Such- und Ersetzoperationen. Um in diesem Text einige Tausend „e" durch „xyz" zu ersetzen, brauchte STeve 45 Sekunden. Die Hälfte des Textes (20.000 Zeilen) löschte Steve innerhalb einer Sekunde. 20.000 Wörter ließen wir mit Hilfe der Dateiverwaltungssprache alphabetisch sortieren: nach 5 Minuten Wartezeit hatten wir die erwünschte Liste. Nahezu jede angewählte Funktion wurde in atemberaubender Geschwindigkeit ausgeführt. Hat man das erlebt, merkt man sofort, daß hier ein Assemblerprofi am Werke war, zumal Steve ohne Ausnahme — vom neugestalteten Objektauswahlfenster bis zum Ressource-File — i der schnellen Maschinensprache abgefaßt ist.

Good bye GEM!

Werfen wir zunächst einen Blick auf die *Benutzeroberfläche*. Steve invertiert nach dem Start automatisch den Bildschirm (weiße Buchstaben auf schwarzem Hintergrund) und ein neuer, proportionaler Zeichensatz mit 64 Buchstaben pro Zeile erscheint (Bild 1). Von einem Pull-Down Menü ist zunächst nichts zu sehen. Erst wenn man die Help-Taste drückt, oder mit der Maus in die rechte obere Ecke fährt, taucht die obligate Menüleiste auf. Jeder Menüpunkt wird noch einmal extra angeklickt (so wie beim Flightsimulator von Sublogic), bevor die Menüauswahl endgültig zur Verfügung steht. Wer bisher mit traditionellen GEM-Programmen gearbeitet hat, wird mit dieser Steuerung anfangs Schwierigkeiten bekommen.

Was sich jedoch dem staunenden Anwender dann präsentiert, ist nur vom Feinsten: 17 unterschiedliche Menüs in insgesamt drei umschaltbaren Ebenen, und fast alle Menüpunkte fakultativ per Tastatur aufrufbar. Leider fallen die Accessories weg, aber Steve beherrscht deren obligate Funktionen zum Teil von Hause aus (Rechnen, Bildschirmdunkelschaltung, Funktionstastenbelegung). Beginnen wir jedoch zunächst mit dem *EDItor* und der Textverarbeitung.

Dem Einsteiger fällt als erstes ins Auge, daß der Cursor von Steve an der Maus „klebt". Bei jeder Mausbewegung gibt es eine Verschiebung in die entsprechende Richtung. Das hat seine Vor- und Nachteile. In jedem Fall wird das Bewegen im Text schneller und einfacher, zumal Tastaturbefehle für das Springen wortweise und zum Zeilenanfang bzw. -ende völlig fehlen. Andererseits reicht schon ein kleiner Ruck an der Maus, um Buchstabensalat auf dem Bildschirm zu erzeugen. Weiß man das, wird man darauf achten, der Maus ein sicheres Plätzchen jenseits der Ellenbogen zu verschaffen. Und dann läßt sich mit Steve gut arbeiten.

Weniger gut gelöst scheint uns das Arbeiten mit mehreren Texten zu sein. Es fehlen die traditionellen GEM-Fenster mit Rollbalken und Textnamen. Vielmehr scrollt man durch eine Datei, indem der Cursor an den oberen bzw. unteren Bildschirmrand bewegt und der Mauskopf gedrückt wird. Oder man ruft einen Zeiger auf, den man auf die gewünschte Textstelle schiebt. Bei bis zu

```
 Disk ! Block  Suchen  Druck  Varia + Form  Obank ? Ende  Font
Nur konsequent ist der  Klassenbegriff  mit  einer  Verteilungs
semantik konnotiert. Er regelt die  Verteilung  des  Verteilens,
nämlich die Verteilung der Individuen auf  Einheiten,  "Klassen"
genannt.  Damit  wird  die  Unverteilbarkeit  des   Verteilungs
prozesses zum Ausdruck gebracht,  d.h.  im  Begriff  selbst  di
Möglichkeit eröffnet, mit der Abschaffung von Klassen  auch  di
Verteilung des Verteilens zugunsten selbstbestimmter  oder  "na
türlicher" Ordnungsformen aufzugeben.
```

Bild 1:
Ein ganz normaler Text, aber mit proportionalem Zeichensatz. Für die Absatzenden fehlt ein sichtbares Steuerzeichen.

```
Dies ist der proportionale Zeichensatz mit 71 Buchstaben pro Zeile.
Selbstverständlich können wir auch russisch schreiben:
селбстверстандлицх кчннен вир аухх руссисцх сцхреибен. Дас техт
зуммдест ауф дем Билдсцхирм мит Стеве охне Бентерес. Und normale
Schrift in der nächsten Zeile. Jetzt kursiv und jetzt fett und nun
unterstrichen. Wenn Sie genauer hinschauen, sehen Sie, daß die
Steuerzeichen in sehr kleinen Buchstaben ihre Funktion wiedergeben.
```

Bild 2:
Mitten in der Zeile kann man zwischen russischem und deutschem Zeichensatz umschalten. Man sieht hier auch die Steuerzeichen für verschiedene Textattribute.

zehn (!) verfügbaren Dateien schaltet man mit der Controltaste und zugehöriger Dateinummer in den jeweiligen Text um. Welche Datei man gerade bearbeitet und wie sie heißt, sieht man nicht. Dazu muß man schon unter einem Menüpunkt nachschauen.

Ungewöhnlich aber pfiffig ausgedacht, ist die Belegung der beiden *Maustasten* während des Schreibvorganges: mit dem linken Knopf fügt man Leerzeichen bzw. -zeilen ein, der rechte Mausknopf löscht das Zeichen unter dem Cursor, bzw. am Zeilenende die ganze Zeile. Selbstverständlich stehen immer noch die liebgewonnene Backspace- und Delete-Taste zur Verfügung, und wer einmal zu viel gelöscht hat, kann mit UNDO alles zurückholen.

Desweiteren unterscheidet sich Steve von herkömmlichen Editoren durch seine *zeilenorientierte Arbeitsweise*. Das Löschen mit Backspace gilt zum Beispiel immer nur für die angewählte Zeile. Auch bei den Blockoperationen muß man stets ganze Zeilen markieren, löschen oder verschieben. Das mag verwöhnte Wordplus-Benutzer irritieren, aber, zumal die wichtigsten Blockbefehle auf dem numerischen Ziffernblock fest belegt sind, ergibt sich ein großer Geschwindigkeitsvorteil gegenüber dem herkömmlichen Mäusespiel. Wer die Belegung vergessen hat, kann natürlich auch im Pull-Down Menü die Verschiebeoperationen anwählen.

Apropos *Ziffernblock*: für ihn gilt in der Regel eine Sonderbelegung mit den wichtigsten und am häufigsten verwendeten Kommandos. Dazu gehören nicht nur die Blockoperationen, sondern auch das Suchkommando oder das Blättern im Text. Insgesamt gesehen eine praktische und schnelle Angelegenheit. Wer dennoch umfangreiche Zahlenkolonnen eingeben möchte, kann natürlich auch die normale Standardbelegung wählen.

Neben diesen obligaten Editorfähigkeiten, bietet Steve einige Schmankerl, die eher exotisch anmuten. Dazu gehört zum Beispiel das Vertauschen von Kleinbuchstaben in Großbuchstaben und umgekehrt. Oder der Wechsel der Schreibrichtung von rechts nach links für arabische Texte. Dazu bedarf es natürlich eines zusätzlichen Zeichensatzes, den man, wie bei Signum, selbst erstellen und verändern kann. Schließlich bietet Steve schon von Hause aus 9 *Schrifttypen* für den Bildschirm: von 40 bis sage und schreibe 160 Zeichen pro Zeile. Schwierig wird es erst bei der Druckerausgabe. Die unterschiedlichen Zeichenbreiten auf dem Schirm ändern an der gedruckten Fassung nichts. Sie dienen nur dazu, eventuell auch bei größeren Tabellen den Überblick zu behalten. Zeilenlineale, wie bei Wordplus, für den Wechsel der Schriftbreite innerhalb eines Dokuments, kennt Steve nämlich (noch) nicht. Allerdings können die *Sonderzeichensätze* (wie der mitgelieferte kyrillischen Font in den Bildern 2 und 3) auch ausgedruckt werden. Dazu muß man allerdings jedes einzelne Zeichen in eine „Download-Tabelle" für den Drucker packen.

Das ist bei einem 24 Nadeldrucker (110 Byte pro Zeichen) sehr viel Arbeit. *Textattribute* (fett, unterstrichen, kursiv, hoch- und tiefgestellt) werden übrigens per Steuerzeichen ein- und ausgeschaltet. Im Gegensatz zu Wordplus ist es damit möglich, alle Attribute über die Suchfunktion zu finden.

Nicht exotisch, sondern einfach und praktisch ist der *Formularmodus*. So ist es problemlos, mit Hilfe des Grafikteils eine ansprechende und übersichtliche Vorlage zu erstellen, wie man beispielhaft in Bild 4 sieht. Wenn man mit Formularen arbeitet, springt der Cursor nach Betätigung der Return-Taste automatisch von einem Feld zum nächsten. In die gleiche Kategorie fällt das *Abkürzungsverzeichnis*. Das ist eine beim Systemstart mitgeladene Datei, in der ein Kürzel und der zugehörige Langtext steht. Tippt man das Kürzel und betätigt anschließend CLR-HOME, so wird automatisch die Bedeutung im Text ergänzt, und zwar in Sekundenbruchteilen.

Zeilen- und Seitenumbruch sind bei Steve keine große Prozedur. Den Text bricht man absatz- oder blockweise um, wobei Steve Trennvorschläge macht. Es existiert kein Trennalgorithmus, so daß die Vorschläge nur per Zufall korrekt sein können. Beim Seitenumbruch gibt das Programm an, wo die Seite normalerweise zu Ende wäre. Will man Absätze nicht auseinanderreißen, schiebt man einen markierten Bereich mit der Maus nach oben. Wenn der Anwender mit verschiedenen Zeilenabständen arbeitet, wird dies beim Seitenumbruch berücksichtigt.

Wichtig für ein neues Paket ist der *Import von Fremdformaten*. Daß Steve als integriertes Programm seine eigenen

Disk | Block Suchen Druck Varia ♦ Form Obank ? Ende Font
selbstachtung
selbständig
selbständigkeit
selbstauslöser
selbstbefriedigung
selbstbehauptung
selbstbeherrschung
selbstbeteiligung
Über STEVE
Einstellung
Maus
Systemmeldungen
Zahlenblock
Direkte Befehle
Grafik
Erweiterte Grafik
ZEICHEN WÄHLEN (MAUSKLICK) UND ÜBERNEHMEN (KLICK)

Bild 3:
Hier sieht man den kompletten Zeichensatz.

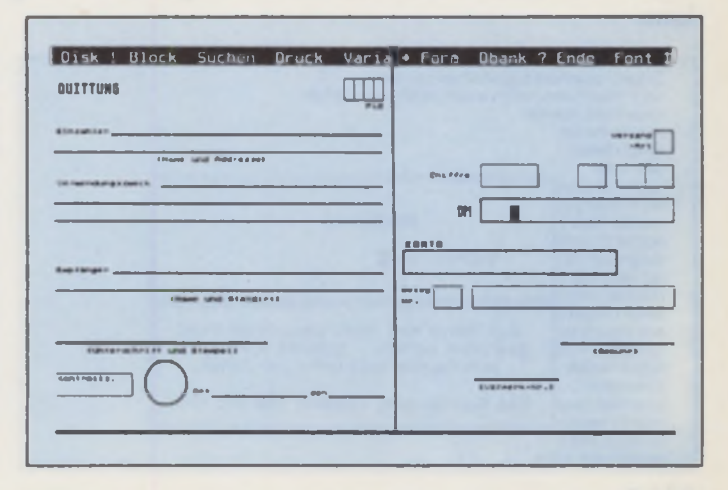


Bild 4:
Formulare zu programmieren, ist eine der Stärken von Steve.

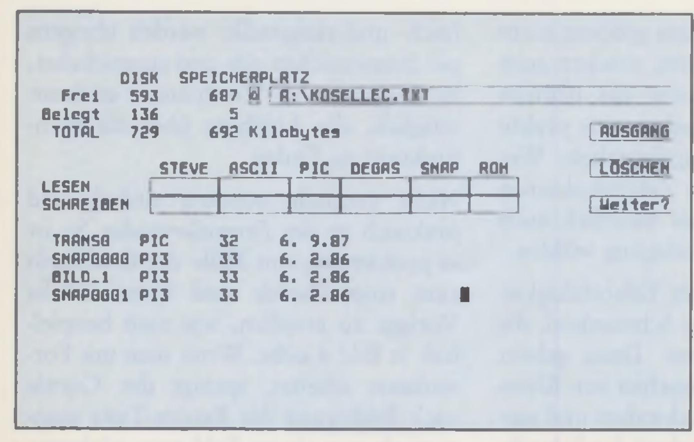


Bild 5:
Das erweiterte Diskmenü zeigt, welche Fremdformate verarbeitet werden.

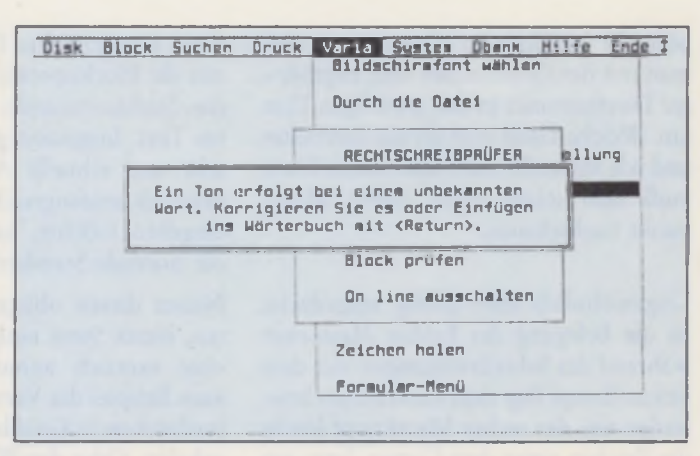


Bild 6:
Der mitgelieferte Spelling Checker umfaßt rund 40.000 Wörter.

Graphiken und Informationen aus der Datenbank im Textmodus verarbeiten kann, ist klar. Wie sieht es aber mit anderen Formaten aus? Im erweiterten Diskmenü (Bild 5) sehen wir, daß Steve auch mit ASCII-Texten, Degas-, Snapshot- und Doodle-Graphiken arbeitet. Der letzte Eintrag („ROH“) ist dazu da, Dateien ohne Modifikation einzuladen und zu speichern. Dabei werden je 50 Byte (gleich welcher Code) mit seitlicher Nummerierung angezeigt, — gleich einem kleinen Diskettenmonitor.

Ein vorläufiges Fazit zur Textverarbeitung: Angenehm überrascht waren wir von der enormen Geschwindigkeit, die Steve in allen Menüpunkten bietet. Etwas umständlich erscheint die Benutzeroberfläche. Unbedingt sollte das Programm um verschiedene Zeilenlineale und eine komfortable Fußnotenverwaltung ergänzt werden; verbessern könnte man den Randausgleich bei Schmalschrift. Schlichtweg ärgerlich ist, daß man Absätze nicht als Ganze einrücken kann.

Mehr als ein normaler Spelling Checker . . .

Neben einer Textverarbeitung, die viele Funktionen in rasantem Tempo erledigt, zeichnet sich Steve auch durch einen Spelling Checker aus, der mit zu den Besten gehört, die für den Atari ST verfügbar sind. Das verwendete Prinzip ist genial. Das Lexikon, mit dessen Hilfe ein Text auf Rechtschreibfehler untersucht wird, ist nämlich nichts anderes, als eine sehr lange Textdatei. Diese wird in den Speicher Nr. 10 eingeladen und kann von jedem anderen Speicherplatz aus aktiviert werden (Bild 6). Man hat die Möglichkeit, entweder einen markierten Block oder den gesamten Text überprüfen zu lassen. Findet Steve ein fehlerhaftes Wort, so wird dieses mit „qq“ markiert. Mit Hilfe der Suchoperation bereitet es dann keine Schwierigkeiten, alle markierten Wörter nacheinander anzuschauen. Selbst bei sehr langen Texten dauerte die Rechtschreibkontrolle in der Regel nur wenige Sekunden: Steve braucht nur die Texte von zwei Arbeitsspeichern zu vergleichen.

Eine zweite Möglichkeit ist die „Online“-Überprüfung. In diesem Fall ertönt noch während des Schreibvorganges ein Signal bei jedem unbekannten Wort, das man dann verbessert oder — mit der Taste Return — neu ins Wörterbuch aufnimmt. Dieses Wörterbuch wird immer komprimiert abgespeichert (rd. 20—30 % Ersparnis), während man bei normalen Texten angeben kann, ob Diskettenspeicherplatz gespart werden soll, oder nicht.

Eine für Textprogramme ungewöhnliche Option hängt eng mit dem Korrekturmodus zusammen. Es ist die *Übersetzungshilfe*. Auf der Systemdiskette wird ein kleines Wörterbuch Englisch-Deutsch mitgeliefert, in dem gesuchte Einträge mit Hilfe der „binären Suche“ recht schnell zu finden sind. Dazu ein Beispiel. Beim Übersetzen wird die Bedeutung des englischen Wortes „mutual“ gesucht. Ein Druck auf Funktions-

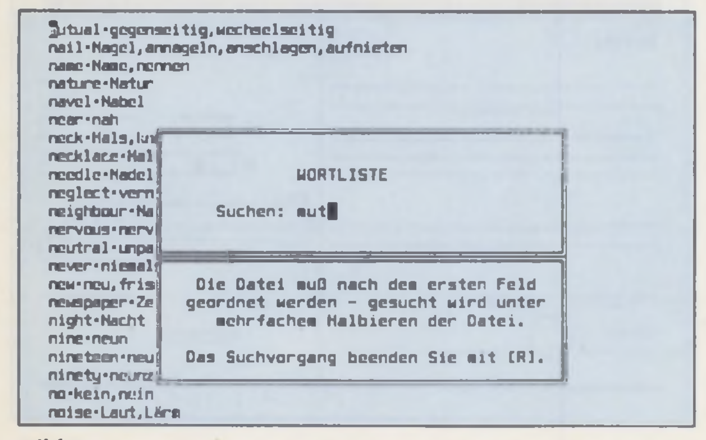


Bild 7:
Die Übersetzungshilfe kann manchen Griff zum Lexikon ersparen.

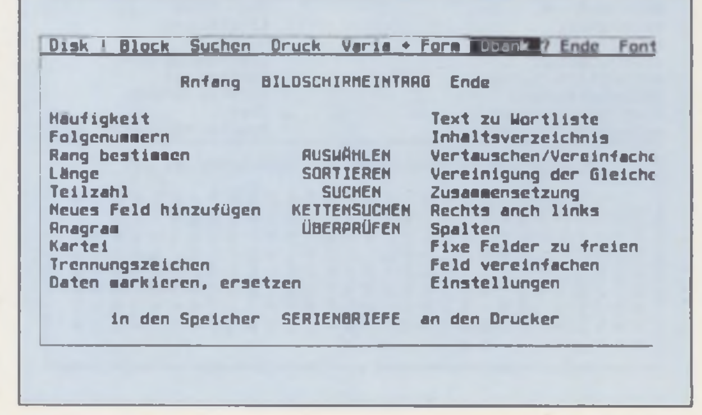


Bild 8:
Alle Datenbankbefehle im Überblick.

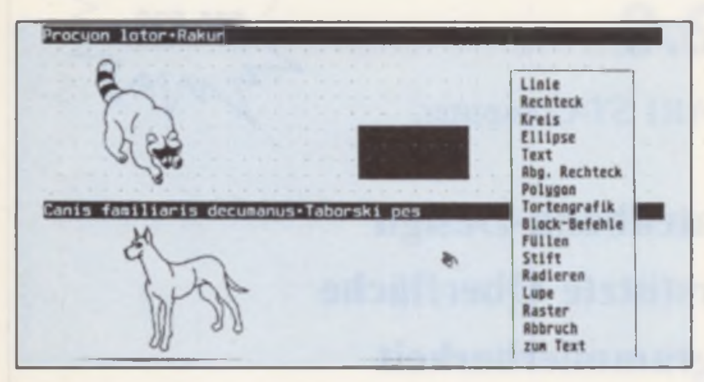


Bild 9:
Bilder in die Dateiverwaltung zu integrieren, ist kein Problem. Die Graphikbefehle werden in einer gesonderten Menüleiste angezeigt.

```
Key: Popper 1957a
Autor: Popper, Karl R.
Titel: Die offene Gesellschaft und ihre Freunde. Berlin, Wien,
Stuttgart, Mainz 1957
Reihe: (Philosophie in Praxis und Gegenwart; 10)
Verlag: Meier & Dunckwarth
Standort: 14 DC 110 P 232    120/34098343
Erstjahr:
Projekt: Wissenschaftstheorie
Oberbegriff: Allgemeine Philosophie
Stichworte: Hegel, Marx
```

Bild 10:
Dies ist ein Dateieintrag mit Maske . . .

taste F9 liefert das Fenster, das in Bild 7 sichtbar ist. Wir tippen „mutual" ein, und noch während wir am Eintippen sind, beginnt Steve mit der Suche. Nach kurzer Zeit sehen wir den englischen Eintrag und die deutsche Übersetzung. Mit der Taste F1 schalten wir in den gerade bearbeiteten Text um und fügen das deutsche Wort ein. Übersetzen (und das heißt ja auch: das passende Wort auswählen) muß man immer noch allein. Aber dennoch würde ein komplettes Wörterbuch auf Diskette unzählige Griffe zum Lexikon ersparen. Eine vorbildliche Idee!

Fließender Übergang: vom Textprogramm zur Dateiverwaltung

Hört man das Wort „Dateiverwaltung", so denkt man gewöhnlich an große Datenmassen, die den Computerspeicher übertreffen, und die man deshalb auf Diskette oder Festplatte auslagert. Dies ist bei Steve nicht so. Vielmehr ist jede Datei im Grunde genommen ein Text, dessen einzelne Datensätze mit Hilfe einer Maske angezeigt werden. Zusätzlich stellt Steve Befehle bereit, mit denen man die Daten sortiert, suchen und darstellen kann, — so wie bei einer Dateiverwaltung (Bild 8). Für den Fall, daß die Datensammlung nicht mehr in den Speicher paßt, darf man auch auf ausgelagerte Dateien zurückgreifen.

Konkret sieht die Verwaltung von Dateien so aus, daß der Anwender zunächst eine Maske für die Datenerfassung erstellt, d.h. bis zu 500 Feldnamen vorgibt. Besondere Angaben über die Feldlänge oder den Feldtyp sind nicht erforderlich, zumal Steve nur Textfelder mit dynamischer Länge (die Feldlänge paßt sich dem eingegebenen Text an) und Bilder verarbeitet (Bild 9). Im Maskenmodus erscheint jeder Datensatz wie auf einer Karteikarte (Bild 10). Gespeichert wird die gesamte Datei jedoch wie ein Text, allerdings mit Trennzeichen zwischen den Einträgen (Bild 11). Für Suchvorgänge stehen die logischen Operatoren NICHT, UND, ODER, GLEICH, GRÖSSER, KLEINER und ENTHÄLT zur Verfügung. Desweiteren natürlich auch Befehle zum Anschauen der Datei: Springen an den Anfang oder das Ende des Files, Blättern im Datensatz, erster und letzter, nächster und vorheriger Satz.

Um den Überblick zu behalten, empfiehlt es sich, die Datei regelmäßig zu *sortieren*. Bis zu 5 Datenfelder kann man gleichzeitig auf- oder absteigend sortieren lassen. Wir haben schon oben auf die rasante Geschwindigkeit verwiesen, in der dies geschieht. Bei einem normalen Atari 1040 ST kann man etwa 75.000 Sätze ohne Probleme sortieren (20 Minuten); bei größeren Datenmassen empfiehlt sich der Einsatz einer Festplatte, um den überforderten Computerspeicher zu entlasten. Für noch größere Datenmengen rät das Handbuch: „transferiere die Datenbank zu einem Mainframe-Computer, sortiere dort und dann zurück . . .".

```
Popper 1957a·Popper, Karl R.·Die offene Gesellschaft und ihre
Freunde, Berlin, Wien, Stuttgart, Mainz 1957·(Philosophie in
Praxis und Gegenwart; 10)·Meier & Dunckwarth·14 DC 110 P 232··
Wissenschaftstheorie·Allgemeine Philosophie·Hegel, Marx
Luhmann 1987b·Luhmann, Niklas·Schreiben als Passion, Wiesbaden
1987··Wissenschaftliche Buchgesellschaft·12 DC L 325··
Soziologie·Systemtheorie·Luhmann, Wissenschaftstheorie
Kuhn 1967·Kuhn, Thomas S.·Die Struktur wissenschaftlicher
Revolutionen. Mit Beispielen aus der Praxis, Frankfurt/M. 1967·
(edition suhrmann; 123)·Suhrmann·Ulrikes Bücherschrank,
zweites Regal··Wissenschaftstheorie··
*********
```

Bild 11:
und in der endgültigen Textform.

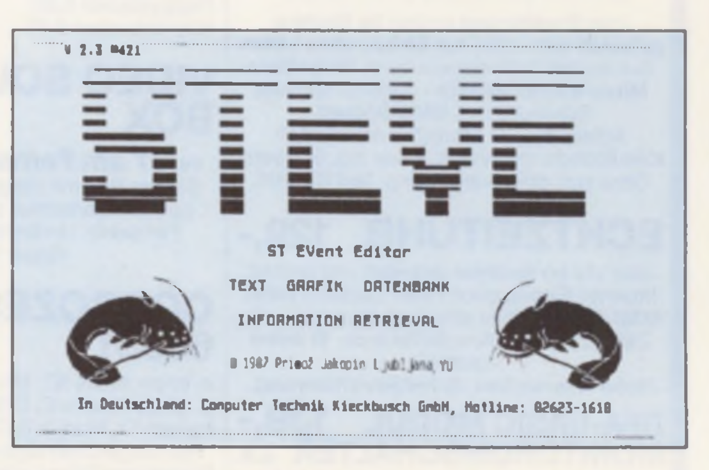


Bild 12:
Das Titelbild — selbstverständlich mit Steve erstellt.



JETZT SPIELEN WIR ALLE TRÜMPFE AUS.
GEMAUERT WIRD NICHT!
EPROM-Programmiergerät:
programmiert EPROMs der 27er Serie von 2764 bis 27256 sowie die entsprechenden CMOS-Typen. Die Einstellung sämtlicher Spannungen erfolgt softwaremäßig. Komplett mit GEM-Software.
DM 249,-
EPROM-Karte 128 KB
128 KB Speicherkapazität. Per Schalter ein- und ausschaltbar. Industriell gefertigt. Mit optischer Zustandsanzeige. Ausführliche deutsche Anleitung.
DM 59,-
Sound-Sampler
10 Bit (!), Musik- und Sprachausgabe bis zu 15 Minuten (bei 1 MB RAM), inkl. 3 Software-Paketen, mit allen Anschlußkabeln sowie seperatem Netzteil, Ausführliches deutsches Handbuch
DM 398,-
jetzt neues Design
Der Super - Trumpf:
eine scharfe Sache für Ihren Fernseher
PAL - INTERFACE III
· ein Feuerwerk der Farben, Darstellung aller 512 möglichen Farben an jedem Farbfernseher.
· jetzt auch Mischfarbendarstellung uneingeschränkt möglich.
· seperate Ausgänge für:
· Antennenbuchse des Fernsehers
· Audioeingang der Stereoanlage
· Videoeingang des Monochrom-monitors/Videorecorders
· lauffähig an allen ST-Versionen
· Lieferung komplett anschlußfertig mit allen Kabeln sowie seperatem Netzteil
· ausführliche deutsche Anleitung.
nur DM 298,-
NEUE PREISE
Die Winteroffensive hat begonnen!
Uhr-Modul
Ausgabe von Datum und Uhrzeit jederzeit möglich. Daten bleiben nach dem Ausschalten des Rechners durch Akkupufferung erhalten. Keine Löt- oder Schraubarbeiten erforderlich. Ausführliches deutsches Handbuch.
DM 99,-
Neu: Monitor-Umschaltung
Monochrom/Farbe
Endlich: Kein lästiges Umstecken mehr. Stecker leiern nicht mehr so schnell aus.
jetzt nur DM 39,90
Weiteres Zubehör:
Zusatzgehäuse aus Metall DM 148,00
Monitor-Verlängerungs-Kabel DM 55,00
Floppy-Verlängerungs-Kabel DM 49,00
Druckerkabel ATARI/Centr. DM 39,00
Monochrom-Monitor-Adapter DM 55,00
Diskettenbox für 50 Disketten DM 29,90
Staubschutzhaube, tranparent, rauchglasfarben DM 29,90
Experimentierplatine für den Expansionsport DM 19,90
Disketten 3,5", 100% Error free, single sided, pro Stück DM 3,40
EPROM 2764 (8 KB) DM 6,90
EPROM 27128 (16 KB) DM 8,90
EPROM 27256 (32 KB) DM 9,90
Joystick 'Competition Pro' mit Mikroschaltern DM 49,00
EPROM-Löschgerät für bis zu 12 EPROMs gleichzeitig DM 115,00
Floppy-Stecker DM 9,90
Monitor-Stecker DM 8,50
und vieles mehr...
Digitalis
Vier Geräte in einem: Oszilloskop, Oszillograph, 10-BIT-A/D-Wandler und Speicheroszilloskop. 86 kHz Abtastfrequenz. GEM-Software, ausführliches deutsches Handbuch.
DM 298,-
Händleranfragen erwünscht.
Achtung: Neue Adresse!
Computertechnik
Z. Zaporowski
Dreieckstr. 2b
5800 Hagen 1
Tel. 02331 / 86555
Vertrieb in der Schweiz
MFS. Sägesser
CH-31 85 Schmitten
Tel. 037 / 36 20 60
Fordern Sie unseren Gratiskatalog an!

```
Escape sequence table - NLQ
Download table
*EpsLQ800,NECP6draft
Epson LQ 800, NEC P6 - draft
Character mapping table - draft
Character mapping table - NLQ
Escape sequence table - draft
Escape sequence table - NLQ
Download table
*Eps LQ800,NECP6 NLQ
Epson LQ 800, NEC P6 - NLQ
Character mapping table - draft
Character mapping table - NLQ
Escape sequence table - draft
Escape sequence table - NLQ
Download table
..
..
```

Bild 13:
Die Druckeranpassung ist nicht sehr übersichtlich.

```
0122142600 w 5
  Datei ausdrucken         1

  Block ausdrucken         2

  Drucker wählen           3

  Drucker einstellen       4

  Druckerdatei laden       5
  Druckerdatei ablegen     6

  1. Tabellenzeile laden   7

  Seitenumbruch            8
..
0129213380 w 6
  Durch die Datei                    1

  Bildschirm löschen und warten      2

  Systemeinstellung                  3
```

Bild 14:
Ein Ausschnitt des Ressource Files für Systemmeldungen.

Auch die *Ausgabemöglichkeiten* sind großzügig ausgefallen, etwa so umfangreich, wie bei Adimens ST. Man kann ein Datenfeld in beliebiger oder fester Länge ausgeben, formatierte Übersichtstabellen erstellen, nur bestimmte Felder selektieren und auch die Datensätze durchgehend nummerieren. Sehr ungewöhnlich sind neben diesen Standardbefehlen einige Kommandos, die speziell auf das Texthandling zugeschnitten sind. So kann man etwa mit dem Befehl „Text zu Wortliste" ein ganzes Textfile in eine Datenbankliste umformen. Nach dem Anwählen von „Vereinigung der Gleichen" entfernt Steve alle doppelten Wörter. Schließlich sortiert man diese Liste und erhält ein alphabetisches Verzeichnis aller im Text verwendeten Worte, das nun entweder für ein Stichwortverzeichnis oder aber die Rechtschreibkorrektur eingesetzt werden kann. Schließlich gibt es auch hier einige exotische Kommandos, etwa „Rechts nach Links", mit dem man ein Wort in seiner umgekehrten Schreibweise erhält (für inverse Wörterbücher) oder die Anagramme, bei denen jedes Wort alle Zeichen in alphabetischer Reihenfolge enthält (aus „Computer" wird dann „CEMOPRTU"). Es ist nicht leicht, an die Grenzen des Befehlsvorrates zu stoßen, weil Steve für alle Fälle gerüstet ist. Und daß sich die Dateiverwaltung nicht mit der Bezeichnung „relational" schmücken darf, fällt bei normalen Anwendungen überhaupt nicht auf.

Graphik mit Pluspunkten

Kommen wir nun zur Graphikausstattung von Steve. Das Programm unterscheidet reine Text- und reine Graphikzeilen, wobei letztere durch Betätigung der Escape-Taste erzeugt werden. Im *erweiterten Graphikmodus* stehen nahezu alle Befehle zur Verfügung, die man auch von Degas kennt. Linien, Rechtecke, Kreise, Ellipsen, Polygone, Tortengraphiken, Blockverschiebungen, Hilfsraster, Zeichenstift und Füllmuster wären etwa zu benennen. Bei allen Funktionen gibt es überdies noch zahlreiche Unterpunkte. So kann man zum Beispiel zwischen 10 Liniendicken mit oder ohne Pfeilende auswählen, oder bei den Füllоperationen unter 36 Mustern das Gewünschte anwählen. Die Blockkommandos sind großzügig ausgefallen, selbst das Invertieren oder Vergrößern eines markierten Bereichs stellt keine Schwierigkeiten dar. Nicht jedes Programm bietet ein unterlegbares Raster zum exakten Zeichnen oder aber Koordinatenangaben zum präzisen Bemaßen, — wohl aber Steve.

Auch der *einfache Graphikmodus* ist nicht ohne Pfiff. Er ist in erster Linie für die Erstellung besonderer Texteffekte vorgesehen. So kann man beispielsweise eine herkömmlich eingegebene Textzeile vergrößern, invertieren, zentrieren oder spiegeln. Die weißen Streifen, die man im Anfangsbild (Bild 12) sieht, geben den Überschriften eine besondere Note.

Die Gretchenfrage des Graphikteils heißt jedoch: wie sieht es mit dem *Ausdruck* aus? Die Druckeranpassung bei Steve scheint komplizierter zu sein, als etwa bei Wordplus (siehe Bild 13). Glücklicherweise werden derzeit fertige Treiber für alle Epson (kompatiblen) Geräte und den NEC P6/7 mitgeliefert. Ansonsten reicht es aus, den linken Rand, einen Seitennummeroffset und Einzel- oder Endlospapier einzustellen. Für Graphiken verlangt Steve eine Eingabe bei „Graphikdichte mit ESC". Gemeint ist der jeweilige Druckermodus mit unterschiedlich vielen Punkten pro Zoll. Auf dem NEC P6 konnten wir hervorragende Ausdrucke im Modus 38 erreichen (180 Punkte pro Zoll), zumal Steve nur mit proportionalen Schrifttypen arbeitet, sah auch der Text mit den frei gestaltbaren Kopf- und Fußzeilen fantastisch aus, — ungefähr so, als würde man mit Wordplus und 1st Proportional arbeiten.

Last not least — ein frei gestaltbares Ressource-Set

Die freilich ungewöhnlichste Eigenschaft von Steve ist das frei gestaltbare Ressource-File. Das heißt für den Anwender, daß er sich eine individuelle Benutzeroberfläche gestalten kann: so mag mancher die Graphikbefehle überflüssig finden und konstruiert eine neue Menüleiste, die nur diejenigen Befehle enthält, die auch wirklich gebraucht werden. Man sucht sich die entsprechende Systemtabelle im Menü „Varia" heraus, verändert sie mit den Textbefehlen und speichert sie ab. In Bild 14 sieht man dazu ein Beispiel. Die erste Zahlenreihe gibt an, an welcher Stelle die Meldung auf dem Bildschirm erscheint, wie groß die dazugehörige Box ist und um welchen Meldungstyp es sich handelt. Zusätzlich kann man bestimmen, was der Benutzer machen muß, um Menüpunkte auszuwählen, oder das Fenster zu verlassen. Kurzum: es handelt sich um alle Einstellungen, die man mit GEM und dem Ressource Construction Set (RCS) vorgibt. Der kleine Unterschied: das RCS-File ist nachträglich vom Anwender kaum zu verändern, während Steve

eine veränderte Benutzeroberfläche beim nächsten Systemstart gleich mitlädt. Dies gilt natürlich auch für die Funktionstastenbelegung, die Bildschirmmasken der Dateiverwaltung das Abkürzungsverzeichnis und vorprogrammierte Batch-Sequenzen (Bild 15).

Fazit

Wir haben festgestellt, daß Steve auf der einen Seite mehr bietet, als jedes andere Textverarbeitungspaket für den Atari ST, auf der anderen Seite bestehen noch geringfügige Ungereimtheiten und Programmierfehler. Eine gewöhnungsbedürftige Benutzeroberfläche wird Anfängern den Start mit Steve erschweren. Und dennoch: Steve ist von seiner Konzeption und Geschwindigkeit her ohne Zweifel ein revolutionäres Ereignis. Programmieren in Assembler heißt der Weg, den Steve weist. Und wie von Computer Kieckbusch zu vernehmen war, wird die jetzt ausgelieferte Version (2.3) nicht die letzte sein. Steve wird also auch in Zukunft für manche Überraschung sorgen.

(M.S.)

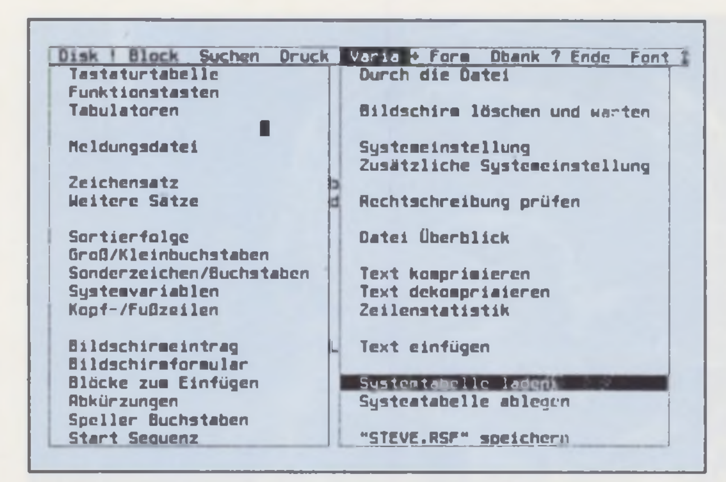


Bild 15: Die Systemtabellen im Überblick — jede einzelne kann individuell angepaßt werden.

Vertrieb:
Computer Technik Kieckbusch GmbH
Am Seeufer 11+12
5412 Ramsbach
Tel.: 02623/1618
Preis: DM 348,—



Elemente der künstlichen Intelligenz

Eine Einführung in Programmiermethoden und Sprachen der KI

7. Teil

Informationsverarbeitung in neuralen Netzwerken

Da wir uns dem Ende dieser Serie nähern, wird es Zeit, sich mit den neueren Methoden der KI und damit auch den Grundlagen menschlicher Intelligenz zu beschäftigen. Denn wie jeder weiß, ist das menschliche Gehirn jedem Computer auf der Welt überlegen, wenn es um sensorische Fähigkeiten und die Fähigkeit zur Abstraktion und Assoziation geht. In diesem Teil möchte ich Sie deshalb mit dem elementaren Aufbau des Gehirns vertraut machen. Und natürlich soll ein Programm die praktische Nutzanwendung (oder einen allerersten Versuch dazu) demonstrieren.

Das Neuron: Das Gatter des biologischen Computers

Basis jeglicher nervlicher Aktivität im menschlichen Körper ist eine spezialisierte Form von Körperzellen, dem Neuron. Abb. 1 zeigt eine schematische Darstellung eines Neurons. Die eigentliche Zelle enthält eine Vielzahl dünner, verästelter Ausläufer, die sogenannten Dendriten. Diese stellen den Kontakt zu benachbarten Neuronen her. Auffällig ist der lange Fortsatz der Zelle, das sogenannte Axon. Dieses bis zu einem Meter lange Axon leitet einen Impuls der Neuronenzelle an die Dendriten benachbarter Neuronen weiter. Über dieses lange Axon (auch unter dem Begriff Nervenfaser bekannt), erfolgt beispielsweise die Mitteilung an das Schmerzzentrum im Gehirn. Uns interessiert hier natürlich weniger die Nervenleitung, als die Wirkung eines neuralen Netzes, eines Ganglions. Wer sich näher für die Funktionsweise eines Neurons interessiert, findet in [1] und [2] leicht verständliche Darstellungen.

Diese Neuronen bilden Netzwerke. Normalerweise herrscht an der Zellmembran ein Potential von −70mV. D. h., das Innere der Zelle ist negativ geladen gegenüber dem äußeren Teil der Membran. Dies rührt daher, daß in der Membran befindliche Proteine Na+-Ionen aus der Zelle herauspumpen. An den Enden der Dendriten befinden sich Kanäle, die in der Regel geschlossen sind, unter Umständen aber geöffnet werden können und durch die Na+-Ionen im geöffneten Zustand nahezu mühelos in die Zelle gelangen können. Dann steigt das Potential im Innern der Zelle wieder an. Überschreitet das Potential einen bestimmten Grenzwert (ca. +20mV), dann bricht ein Chaos aus! Das Neuron pumpt Na+-Ionen in das Axon, so daß sich über die Membran des Axons eine Depolarisierung ausbreitet. Das Neuron feuert einen Impuls ab. Der zeitliche Verlauf dieses Aktionspotentials ist in Abb. 2 dargestellt. Kurz nach dem Feuern des Neurons braucht es eine Ruhephase, in der kein weiterer Impuls abgefeuert werden kann (absolute Ruhephase).

Daran schließt sich eine relative Ruhephase an, in der eine Anregung manchmal zum Abfeuern führt, manchmal aber auch nicht (relative Ruhephase). Erst danach hat sich das Neuron erholt und reagiert wie gewohnt. Am Ende des Axons befindet sich wieder ein Dendrit eines Neurons, das wiederum angeregt wird. Manchmal gelangen Substanzen durch die Kanäle der Synapsen (Stelle an der Axon des anregenden Neurons und Dendrit des angeregten Neurons zusammentreffen), die das Potential im Innern der Zelle noch negativer werden lassen. In diesem Fall wirkt das Neuron beim Feuern nicht anregend (excitatorisch), sondern verhindernd (inhibitorisch). Weiterhin ist interessant, wo das Neuron angeregt wird.

Je näher der Dendrit am Axon sitzt, desto stärker wirkt sich das Einströmen von K+ −Ionen auf das Axon aus, d. h. das Neuron erfährt einen stärkeren Potentialanstieg. Außerdem



können sich benachbarte Neuronen bei Anregung gegenseitig verstärken, da im gleichen Raumgebiet eine lokale Erhöhung der Na+-Konzentration erfolgt. In Wirklichkeit laufen diese Vorgänge natürlich sehr komplex ab. Erstens hat ein Neuron hunderte von Dendriten, zweitens kann sich ein Neuron auch selbst erregen (über einen Ausläufer des Axons an einen eigenen Dendriten). Um aber einen Einblick in den Mechanismus eines Ganglions zu bekommen, habe ich das Programm GANGLION in die PD gegeben. Es ist in C geschrieben (Megamax) und simuliert ein Ganglion, bestehend aus 8 peripheren Neuronen und einem Zentralneuron. Beim Starten des Programms werden die Neuronen per Zufallsgenerator auf Excitatorisch oder Inhibitorisch eingestellt. Abb. 3 zeigt das Bild nach dem Einschalten. Die Wirkung eines Neurons kann mit Hilfe des Menüpunktes **'Wirkung'** der Menüleiste geändert werden. Man klickt den Menüpunkt an und anschließend das Neuron, dessen Wirkung geändert werden soll. Ansonsten klickt man das Neuron an, das gezündet werden soll. Hat das Neuron gezündet, erkennt man das einerseits daran, daß der Inhalt invertiert wird, andererseits steigt das Potential im Innern des Neurons. Wird ein 'excitatives Neuron' dicht am Axon gezündet, wird das Neurons sofort feuern. Ist die Wirkung des Neurons 'inhibitorisch', fällt das Potential unter −70mV und verhindert damit ein Zünden bei der nächsten 'excitatorischen' Anregung. Das gezündete Neuron bleibt für die Dauer der absoluten Erholungsphase invertiert dargestellt. Der Leser möge mit dem Programm spielen und die Zusammenhänge, die (unendlich komplexe) Grundlage unseres Denkens darstellen und auf sich wirken lassen.

Abb. 1: Schematische Darstellung eines Neurons

Die Matrix: Das biologische Speichermedium mit mathematischen Hintergrund

Nachdem sie nun gesehen haben, wie ein Netzwerk von Neuronen zusammenwirkt, bleibt die Frage, wie dieser physiologische Aufbau zum Denken, d. h. zum Lernen und Speichern von Informationen benutzt werden kann. Fassen wir das Ergebnis des vorigen Abschnitts nochmals in Abb. 4 zusammen. Die Synapsen an den Dendriten eines Neurons erhalten Signale von dem Axon eines verbundenen Neurons. Im Innern der Zelle werden die Signale der Neuronen addiert. Das Neuron feuert, wenn ein Schwellwert überschritten wird. Die miteinander verbundenen Neuronen bilden ein Netzwerk, wie es in sehr einfacher Form Abb. 5 zeigt.

Gestrichelt sind dort die inhibitorisch wirkenden Verbindungen eingezeichnet und solide die exhibitorisch wirkenden Verbindungen. Wie man sieht, läßt sich das gleiche Netzwerk in graphischer und in Matrizenform darstellen. In der Letzteren wird eine exhibitorische Verbindung als +1 an der entsprechenden Stelle eingetragen, eine inhibitorische Verbindung als −1 und keine Verbindung als 0. Mit Hilfe solch einer Matrix ließe sich das Verhalten eines einfachen Neuronenkomplexes bereits simulieren. Aber wo steckt die Information?

Digitale Mustererkennung

Besonders anschaulich ist der Prozeß des Lernens und Erinnerns von Informationen am Beispiel der Mustererkennung. Bekanntlich erhalten wir unser Sehvermögen durch eine Vielzahl von Sehnerven, die in der Netzhaut ausliegen und durch Licht angeregt werden. In unserem einfachen Neuronenmodell entspricht das einer rechteckigen Anordnung von Neuronen.

In Abb. 6 sind neun Neuronen gezeigt, die ein Bild des Buchstabens I erhalten. Für eine brauchbare Mustererkennung sind mehr als nur 9 Neuronen erforderlich, aber es wird auch so schon kompliziert genug. Die Information ob ein Neuron belichtet wird oder nicht, ist in dem Zustandsvektor V gespeichert. Eine 1 bedeutet, daß ein Teil des Buchstabens über dem Neuron liegt. Eine 0 bedeutet kein Signal für das Neuron. Wenn nun ein Bildwechsel erfolgt, soll sich das Gedächtnis wieder erinnern, wenn erneut ein I als Eingangssignal vorliegt, d. h. der Zustandsvektor identisch ist.

Wie wir aus dem oberen Abschnitt noch wissen, ist ein Neuron auf Grund seiner Dendriten und axonalen Verzweigungen in der Lage, mit jedem anderen Neuron eine Verbindung herzustellen. Die Information, wie eine Verbindung wirkt, stellen wir wieder als 1 (excitatorisch) oder −1 (inhibitorisch) dar. Und wie die Wirkung sein soll, bestimmen wir aus dem Zustandsvektor in der Lernphase. Es gibt verschiedene Lernregeln, nach denen der Wert der Lernmatrix berechnet werden kann. Wir verwenden in dem Programm NEURAL.PAS (Listing 1) ausschließlich die Hopfieldschen Lernregel. Wer sich stärker für dieses Thema interessiert und einen einführenden Übersichtsartikel über Wissensspeicherung

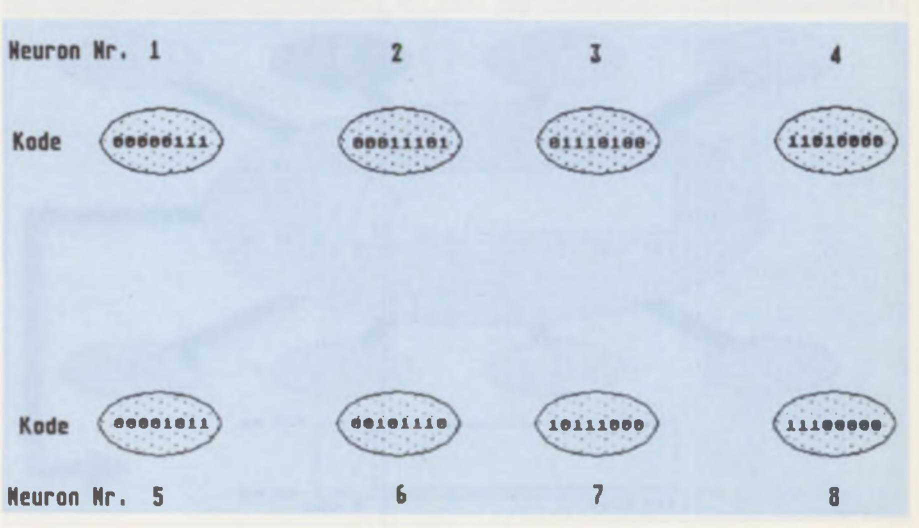


Abb. 2: Der zeitliche Verlauf eines Aktionspotentials



in neuralen Netzen sucht, sei an [3] und [4] verwiesen.

Nach der Hopfieldschen Lernregel berechnet sich die Zustandsmatrix aus deren vorliegendem Wert und dem Zustandsvektor:

Gl.1

$$W_{ij} = W_{ij} + (2 \star V_i - 1) \star (2 \star V_j - 1)$$

Mit anderen Worten sagt Gl.1: der neue Wert der Matrix in der Spalte i und der Zeile j ist gleich dem alten Wert zuzüglich +1 oder −1. Und zwar wird 1 addiert, wenn die i-te Stelle und die j-te Stelle des Zustandsvektors V entweder beide 0 oder beide 1 sind. Haben beide Stellen des Zustandsvektors verschiedene Werte, so wird 1 subtrahiert. Damit ist klar, daß die Matrixelemente dort betragsmäßig erhöht werden, wo oft dasselbe Muster erscheint. Das Muster prägt sich der Matrix langsam ein, der Lernprozeß hat begonnen.

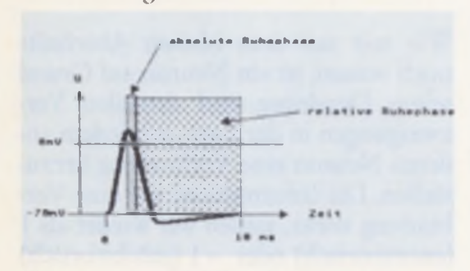
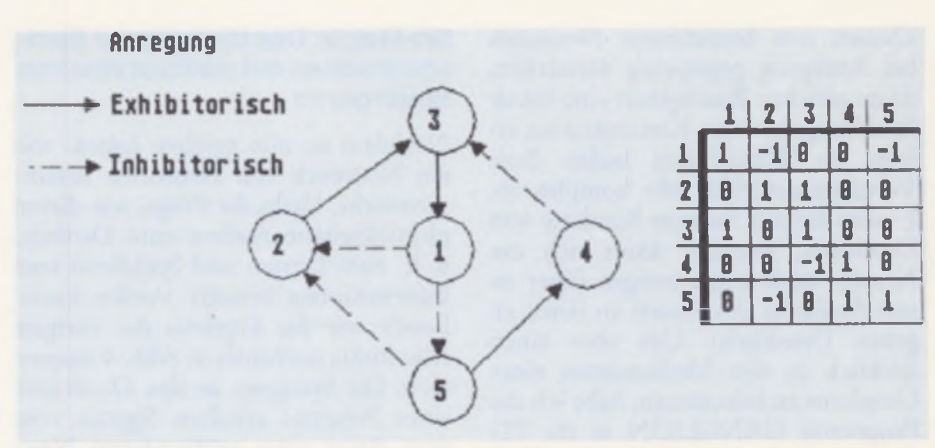


Abb. 3: Potentialverlauf nach dem Einschalten

Erinnerung an Gelerntes

Nachdem der Matrixspeicher genügend Gelegenheit zum Lernen hatte, überlegen wir uns nun, wie man seine Erinnerung wecken kann. Dazu müssen



	1	2	3	4	5
1	1	-1	0	0	-1
2	0	1	1	0	0
3	1	0	1	0	0
4	0	0	-1	1	0
5	0	-1	0	1	1

Abb. 5

wir den momentan vorliegenden Zustandsvektor mit der Lernmatrix verknüpfen:

Gl.2 $$V' = V \star W$$

In dieser einfachen Form ist auf die Berücksichtigung eines konstanten Eingangssignals und des Schwellwertes der Einfachheit halber verzichtet worden. Hierin bedeutet *V'* den Ausgangszustandsvektor, *V* den Eingangszustandsvektor und *W* die Lernmatrix. Nach den Regeln der Multiplikation einer Matrix mit einem Vektor errechnet sich somit die i-te Komponente des Ausgangszustandsvektors *V'* zu:

Gl.3

$$V'_i = \sum_{j=1}^{9} W_{i,j} * V_j$$

In Abb. 6 ist die Lernmatrix *W* nach einmaligem Lernvorgang für 9 Neuronen, sowie der sich daraus ergebene Zustandsvektor und das entsprechende Muster dargestellt.

Lohnt sich der Aufwand?

Um selbst Erfahrung mit dem Matrixgedächtnis sammeln zu können, habe ich das Programm NEURAL.PAS geschrieben (Listing 1). Es ist mit ALICE Pascal entwickelt, da es so schneller läuft. Für Benutzer anderer Pascal Versionen sollte die Umsetzung keine Schwierigkeiten bereiten. Interessierte ohne Pascal Interpreter oder Compiler finden auf der Programmdiskette dieses Heftes den Run-Only-Interpreter APRUN, mit dessen Hilfe das Programm gestartet werden kann.

Abb. 7 zeigt den Bildschirm nach dem Start des Programms und einer kleinen Copyright Nachricht. Der Fadenkreuzcursor kann jetzt in das Eingabefeld bewegt werden. Durch Anklicken eines Feldes wird das Feld (entspricht einem Neuronenmodell) schwarz dargestellt (Neuron aktiv). Abb. 8 zeigt den Bildschirm nach der Eingabe des Buchstabens A. Natürlich erscheint nun auch das gleiche Muster in der Ausgabe. Wir lassen mit Hilfe mehrerer Doppelklicks die Matrix den Buchstaben lernen. Anschließend geben wir ein **verrauschtes** Exemplar desselben Buchstabens ein und lassen die Ausgabe berechnen. Wie Abb. 9 eindeutig zeigt, erhält man ein fehlerfreies Ergebnis.

Heureka. Ist es nicht gerade diese Eigenschaft, die menschliches Erkennen auszeichnet? Das Erkennen von undeutlichen oder mehrdeutigen Signalen? Es scheint, als wäre man mit dem Matrixgedächtnis dem Geheimnis natürlicher Intelligenz ein ganz klein wenig näher gekommen. Noch etwas ist phantastisch an diesem primitiven Matrixspeicher. Als nächstes geben wir (ohne das Programm zwischendurch gestoppt zu haben) den Buchstaben X

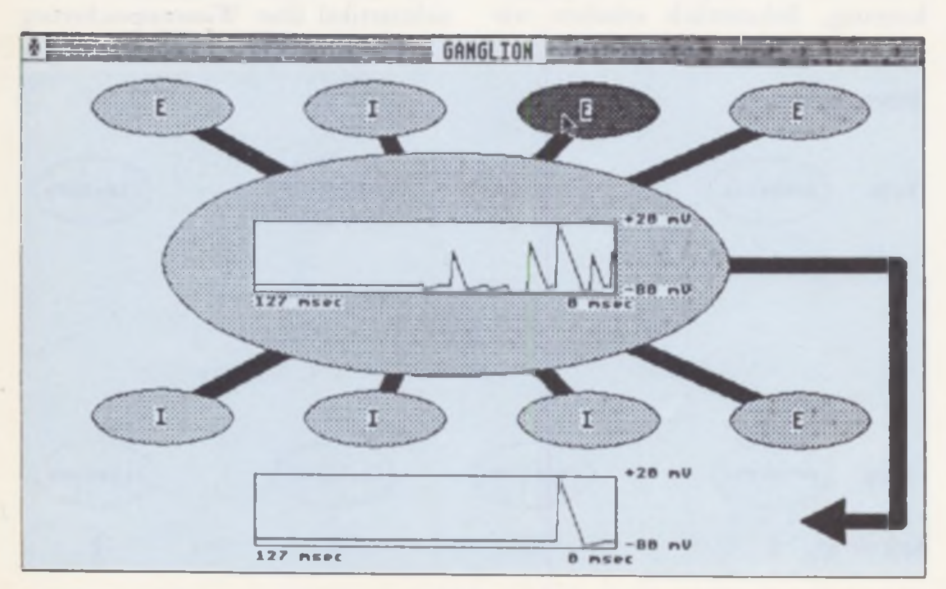


Abb. 4



ein. Abb. 10 zeigt, daß die Matrix den Buchstaben **X zusätzlich** gelernt hat. Überprüfen Sie das gerne, indem Sie nacheinander **X, A, X** etc. eingeben. Immer erhalten Sie die richtige Antwort. Wenn nicht, haben Sie einfach nicht lange genug gelernt (wie menschlich!!). Aber Halt! Nicht gleich übermütig werden! Falls Sie versucht haben sollten, mehr als zwei Buchstaben zu lernen, wird die kleine Matrix Sie leider mit einem wirren Muster enttäuschen müssen (oder nennt man das vielleicht Kreativität?).

Weiterführende Anwendungen

Natürlich läßt sich das Verfahren noch erheblich verbessern. So fehlt beispielsweise ein Eichfaktor, der mit dem Eingangssignal des Neurons multipliziert, die Position der Synapse berücksichtigt. Und auch die Schwellwertfunktion wurde vernachlässigt. Und selbstverständlich ist es nicht der Sinn der Matrixspeicherung, das Bild des Eingangsvektors zu reproduzieren. Vielmehr soll mit dem Vektor anschließend etwas gemacht werden. Beispielsweise die Zuordnung des ASCII Zeichens an das gelesene Muster. Oder im Bereich der Spracherkennung denke man sich das Sprachsignal einer Fourieranalyse unterzogen, und die daraus resultierende Frequenz-Zeit-Matrix wird in einem Matrixspeicher gespeichert und der gesprochene Satz niedergeschrieben. Vielleicht ist dies der Weg zur vollautomatischen Schreibmaschine? Wer sich näher für das assoziative Gedächtnis interessiert, sollte bei [5] nachlesen.

Dr. Sarnow

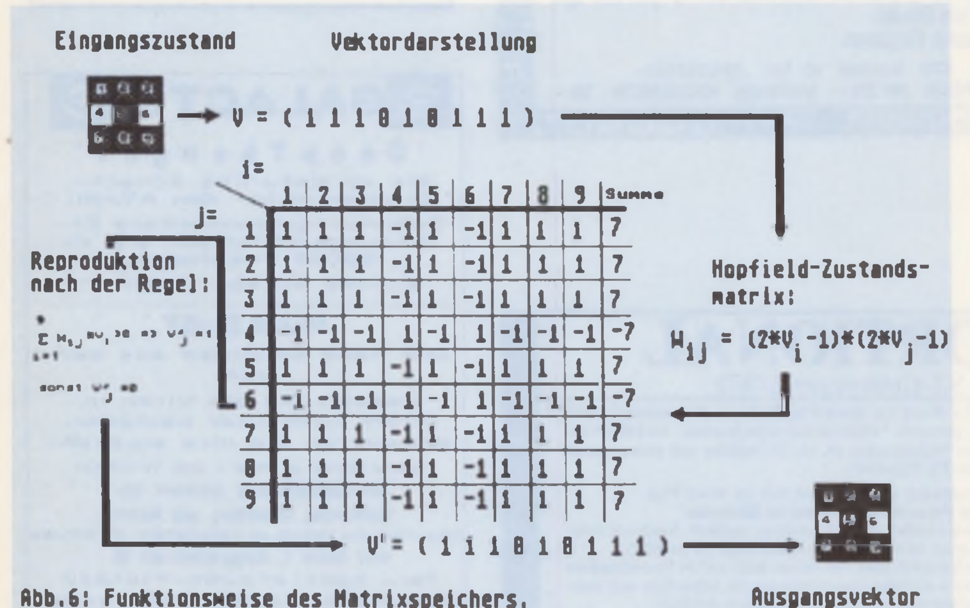


Abb.6: Funktionsweise des Matrixspeichers.

Literatur

[1] B. Douglass, 80 Microcomputing, April 1982, Copernicae Mathematicae, p. 382.

[2] Spektrum der Wissenschaft, Sonderband Gehirn und Nervensystem. Heidelberg 1987.

[3] C. Jorgensen, C. Matheus: Catching Knowledge in Neural Nets. AI Expert, December 1986, p. 30 ff.

[4] S. B. Schreiber, Großer Auftritt für eine kleine Matrix, c't, Juli 1987, p. 106 ff.

[5] T. Kohonen: Self-organisation and associative memory. Springer Verlag, Berlin, 1984.

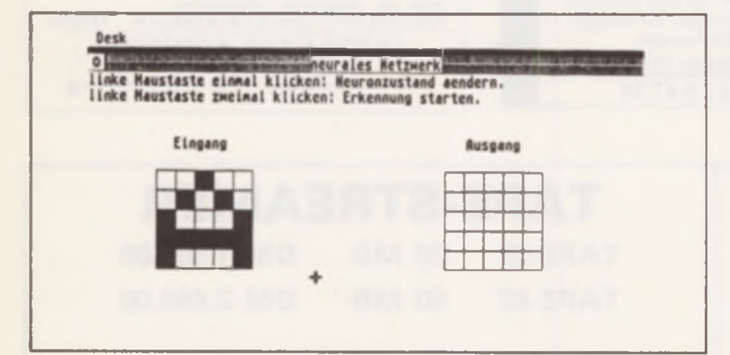


Abb. 7: Der Buchstabe A im Eingangsvektor.

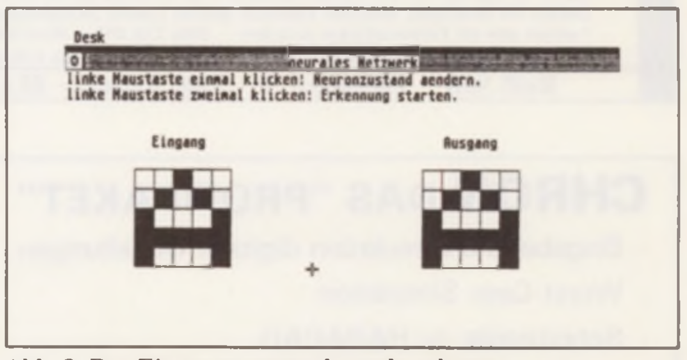


Abb. 8: Der Eingangszustand wurde gelernt.

```
 1:    program netzwerk(input, output);
 2:    (Simulation eines neuralen Netzwerkes)
 3:
 4:    ( ALICE Benutzer MERGEen hier die Bibliothek GRAPHLIB.AP )
 5:    ( Benutzer anderer PASCAL Versionen Includen hier die GEMLIB )
 6:
 7:    const
 8:            anzahl = 25;        (Anzahl Neuronen im Netzwerk)
 9:            zeile = 5;    (Anzahl Neuronen pro Zeile)
10:            py = 100;      (y-koordinate der Rechteckfelder)
11:            pxe = 70;      (x-koordinate des Eingangsfeldes)
12:            pxa = 350;    (x-Koordinate des Ausgangsfeldes)
13:            breite = 20;        (Breite eines Feldes)
14:
15:    type
16:            zustand = array [1..anzahl] of integer;  (Zustaende der einzel
                nen Neuronen)
17:            matrix = array [1..anzahl] of array [1..anzahl] of integer;  (
                Relationsmatrix)
18:
19:    var
```

```
20:            eingang, ausgang : zustand;  (
21:            Eingangssignal des neuralen Netzwerkes und Ausgang auf Grund d
                es neuralen Lernvorganges)
22:            w : matrix;  (Die Relationsmatrix der neuralen Zustaende)
23:            i, j, ereignis, mx, my, neuron_nr : integer;
24:            fenster : text;
25:            menueleiste : Pointer;
26:
27:    procedure hopfield(eingang: zustand; var w: matrix);
28:            (Hopfield'sche Lernregel)
29:            var
30:                    i, j : integer;
31:
32:        begin
33:            for i := 1 to anzahl - 1 do begin
34:                    for j := i + 1 to anzahl do begin
35:                            w[i][j] := w[i][j] + (2*eingang[i] - 1)*(2*ein
                                gang[j] - 1);
36:                            w[j][i] := w[i][j];
37:                        end;
38:                end;
```



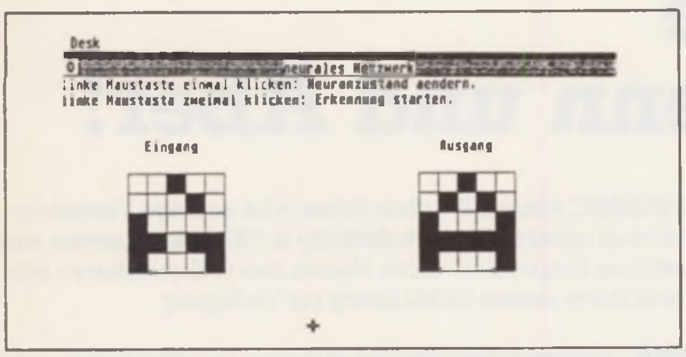


Abb. 9: Erkennung einer unvollständigen Eingabe.

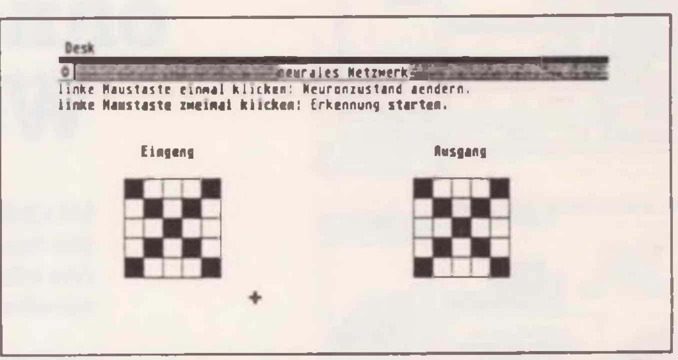


Abb. 10: Die gleiche Matrix kann noch einen weiteren Eingangszustand lernen.

```
 39:        end;
 40:    procedure zeige_neuron(n: integer);
 41:            (Zeigt den Zustand des n-ten Neurons)
 42:            var
 43:                    i, j, k : integer;
 44:                    punkte : array [1..4] of record
 45:                            x, y : integer;
 46:                        end;
 47:
 48:        begin
 49:            i := (n - 1) mod 5;
 50:            j := (n - 1) div 5;
 51:            if eingang[n] > 0 then begin
 52:                    FillPattern(1);
 53:                end
 54:             else begin
 55:                    FillPattern(0);
 56:                end;
 57:            for k := 1 to 4 do begin
 58:                    with punkte[k] do begin
 59:                            x := pxe + i*breite + ((k - 1) mod 2)*breite;
 60:                            y := py + j*breite + (k mod 2)*breite;
 61:                        end;
 62:                end;
 63:            w_bar(punkte);
 64:        end;
 65:    procedure zeig_zustand ;
 66:            (Zeigt den Zustand des Eingangs- und Ausgangsfeldes an)
 67:            var
 68:                    i, j, k : integer;
 69:                    punkte : array [1..4] of record
 70:                            x, y : integer;
 71:                        end;
 72:
 73:        begin
 74:            for i := 1 to zeile do begin
 75:                    for j := 1 to zeile do begin
 76:                            zeige_neuron(i + (j - 1)*5);
 77:                            if ausgang[i + (j - 1)*5] > 0 then begin
 78:                                    FillPattern(1);
 79:                                end
 80:                             else begin
 81:                                    FillPattern(0);
 82:                                end;
 83:                            for k := 1 to 4 do begin
 84:                                    with punkte[k] do begin
 85:                                            x := pxa + i*breite + ((k - 1)
                                               mod 2)*breite;
 86:                                            y := py + (j - 1)*breite + (k
                                               mod 2)*breite;
 87:                                        end;
 88:                                end;
 89:                            w_bar(punkte);
 90:                        end;
 91:                end;
 92:        end;
 93:    procedure neuer_ausgang(var ausgang: zustand; w: matrix);
 94:            {
 95:            Berechnet den neuen Ausgangszustand aus der gelernten Matrix
                und dem Eingangszustand)
 96:            var
 97:                    zwischen : zustand;
 98:                    i, j, summe : integer;
 99:
100:        begin
101:            for i := 1 to anzahl do begin
102:                    summe := 0;
103:                    for j := 1 to anzahl do begin
104:                            summe := summe + w[i][j]*eingang[j];
105:                        end;
106:                    if summe > 0 then begin
107:                            ausgang[i] := 1;
108:                        end
109:                     else begin
110:                            ausgang[i] := 0;
111:                        end;
112:                end;
113:        end;
114:    function finde_neuron(mx, my: integer) : integer;
115:            {
116:            Diese Funktion gibt die Nummer des Neurons (1..25) zurueck,
                 welches von der Maus angeklickt wurde)
117:            var
118:                    i, j : integer;
119:
120:        begin
121:            if ((mx < pxe + 5*breite) and (mx > pxe) and (my > py) and (my
                  < py + 5*breite)) then begin
122:                    i := (mx - pxe) div breite;
123:                    j := (my - py) div breite;
124:                    finde_neuron := j*zeile + i + 1;
125:                end
126:             else begin
127:                    finde_neuron := 0;
128:                end;
129:        end;
130:    (Hauptprogramm)
131:    begin
132:    RemoveEditWindows ;
133:    Close(output);
134:    GemStart ;
135:    menueleiste := NewMenuBar ;
136:    DisplayMenuBar(menueleiste);
137:    Assign(fenster, 'neurales Netzwerk');
138:    QuickWindow(fenster, WOtitle or WOclose or WOclicks, 0);
139:    GraphicsWindow(fenster);
140:    writeln(fenster, 'linke Maustaste einmal klicken: Neuronzustand
         aendern.');
141:    writeln(fenster, 'linke Maustaste zweimal klicken: Erkennung starten.'
         );
142:    writeln(fenster);
143:    writeln(fenster);
144:    writeln(fenster, '          Eingang                                    Aus
         gang');
145:    for i := 1 to anzahl do begin
146:            eingang[i] := 0;
147:            ausgang[i] := 0;
148:            for j := i to anzahl do begin
149:                    w[i][j] := 0;
150:                    w[j][i] := 0;
151:                end;
152:        end;
153:    i := Alert(1, '[1][<c> Dr. Sarnow|geschrieben in|ALICE-Pascal]
         [Alles klar?]');
154:    zeig_zustand ;
155:    MouseType(7);
156:    repeat
157:            ereignis := GetEvent(true);
158:            if ereignis = Eclick then begin
159:                    if EventParameter(0) = 1 then begin
160:                            mx := EventParameter(1);
161:                            my := EventParameter(2);
162:                            neuron_nr := finde_neuron(mx, my);
163:                            if neuron_nr > 0 then begin
164:                                    if eingang[neuron_nr] = 0 then begin
165:                                            eingang[neuron_nr] := 1;
166:                                        end
167:                                     else begin
168:                                            eingang[neuron_nr] := 0;
169:                                        end;
170:                                    zeige_neuron(neuron_nr);
171:                                end;
172:                        end;
173:                    if EventParameter(0) = 2 then begin
174:                            MouseType(2);
175:                            hopfield(eingang, w);
176:                            neuer_ausgang(ausgang, w);
177:                            zeig_zustand ;
178:                            MouseType(7);
179:                        end;
180:                end;
181:    until ereignis = Eclose;
182:    MouseType(0);
183:    GemFinish ;
184:    end.
```

Listing 1: Simulation eines neuralen Netzwerks

Kieckbusch Window

STEVE

348,– DM

Integriertes Programm: Text – Grafik – Datenbank

Der Programm-Knüller der Saison!!!

Programm in Maschinensprache, daher enorm schnell
Text: Online-Spell-Checking
Lexikon für Deutsch, Englisch, Russisch,
Jugoslawisch verfügbar
Übersetzt wörtlich Deutsch – Englisch
8 Schriften, eigener Zeichengenerator
schreibt von links nach recht und umgekehrt
umfangreiche Rechenfunktionen integriert
Übernahme von Grafiken aus fremden Programmen
Serienbriefe mit Adressen aus der Datenbank

*

Grafik: Zeichenmöglichkeiten ähnlich Easy Draw
z. B. Rechteck, Kreis, Ellipse, Polygon, Fill, Pen usw.
verschiedene Zeichenblatt-Größen
Zeichnungs-Übergabe an die Textverarbeitung

*

Datenbank: variable Datensatzlänge, bis 500 Felder/S.
einfache Formblatterstellung, abspeicherbar
sehr schnelles Sortieren: 10.000 Sätze in 1 Min.
über 20 Datenbank-Funktionen
Verwaltung von Textbausteinen

STEVE-Lehrbuch

Weil STEVE so mächtig ist, kann ein zusätzliches
Lehrbuch inkl. einer Beispiel-Disk sehr hilfreich sein.
250 Seiten, 480 Gramm

Preis: 58 DM

Hausverwaltung ST

798,– DM

Ein wirklich professionelles Programm:

100 Objekte mit je bis zu 100 Einheiten
Stammdatenverwaltung, Buchungsroutinen
Listenerstellung: Mieter, Eigentümer, Kosten usw.
Automatische Sollstellung
Automatisches Mahnwesen
Textverarbeitung mit Serienbrief
und vieles Mehr

Netzwerk-Berechnung

Berechnet in komfortabler Weise elektrische Netzwerke!
Sehr schnell und voll unter GEM.

Preis: 98 DM

VIP Professional 1.4

– Kalkulation
– Grafik
– Datenbank

Daten- und Bedienungskompatibel
mit LOTUS 1-2-3
kpl. in deutsch, inkl. Hotline u. Updateservice
748 DM
Mit 3500 verkauften Paketen in Deutschland
das meistbenutzte Kalkulationsprogramm!!

VideoMixer

Synchronisation und Überlagerung von
Computer-Bild und Video-Bild.

Preis: 1.298,– DM

Multi-Hardcopy ..... 98 DM

Anpassung an wirklich jeden Drucker,
ob 8, 16 oder 24 Nadeln, ob Farb- oder Laser-Drucker

Video Construction Set PLUS

Jetzt mit 3D-Editor, 3D-Animation, alle Auflösungen,
mehrere Filme gleichzeitig, automat. Farbanimation,
Zufallsfarben, Lupe, Effekte, Schnittstelle zu GFA,
C, Assembler usw., unterstützt Sound-Digitizer und
Blitter, hat GENLOCK- und CLIP-Modus
und vieles mehr

Preis: 198,– DM

Desk Assist II+

Das Rundumprogramm für Ihren Atari ST

Terminplanung, Kalender, Uhr (auch in der Menüzeile)
Alarmtermine (Anzahl unbegrenzt), Notizblatt
Rechner (dez/hex/bin/Zeit/Datum/
Maßumrechnung/18-stellig)
residente Adressen und Telefondatei
mit komfortablen Such- und Druckmöglichkeiten
Druckerspooler und Hardcopy (auch Teilbild)
ausgefeilte Drucker-Anpassungs-Möglichkeiten
Verschlüsselung beliebiger Dateien
Super-Terminalprogramm im Lieferumfang!!!
zuverlässiges deutsches Produkt

Preis: nur 98,– DM

In diesen progressiven Computer-Häusern finden Sie unsere Programme und die ausführliche Beratung dazu:

OS-9 ist lieferbar!

Paket enthält: OS-9 auf Modul inkl. Hardware-Uhr,
Screen Editor, Macroassembler, Basic, Pascal,
C-Compiler, Text, Mailmerge, Spellchecker,
Spreadsheet, Database,
3 Ordner Dokumentation,
über 1000 Seiten Software-Tool,
Harddisk Tool mit OS-9 Partition, 3 DD Disketten
alles engl., Umlaute o.k.

Preis: 1099 DM

Schluß mit den Spielereien:

Word Perfect

Das Marktführer-Programm aus der IBM-Welt
nun auch auf Atari ST!
Textverarbeitung in höchster Vollendung.
Da bleibt wirklich kein Wunsch offen.
Lieferbar im November.
Hier kann man lange erzählen, was es alles kann.
Fordern Sie unseren ausführlichen Prospekt an.

Preis: 899 DM

Privatliquidation

für Ärzte hilft Ihnen bei:

der Suche nach den Gebührenordnungsziffern
der Synonymensuche und GOÄ-Ziffer in der GOÄ
der Verwaltung der Patienten-Stammdaten
der Erstellung von Privatliquidationen
der Buchführung im Rechnungswesen
der Verwaltung der Labordaten
der Erstellung von Formularen und Arztbriefen
der Kassenabrechnung,
der Anmahnung der Krankenscheine
der Anmahnung der Rechnung,
dem Überblick über Außenstände
Fordern Sie ausführlichen Prospekt an.

Preis: 498 DM

A-MAGIC Turbo Dizer

Das non plus ultra unter den Video Digitizern
Echtzeit-Verarbeitung in 16 Graustufen
Weiterverarbeitung in allen bekannten
Mal- und Zeichenprogrammen
Ein Schweizer Präzisionsgerät für *498,– DM*

1000 Berlin 31	Alpha Computers GmbH
1000 Berlin 30	Computare Elektronik OHG
1000 Berlin 31	Data Play
2000 Hamburg 71	Createam Microcomputer
2000 Norderstedt	Sellhorn GmbH
2080 Pinneberg	B P O GmbH
2300 Kiel 1	MCC Micro Comp. Christ
2400 Lübeck	Jessen & Lenz
2400 Lübeck	Warter
2800 Bremen	PS-Data
2900 Oldenburg	Carl Wölthe GmbH & Co KG
2940 Wilhelmshaven	Radio Tiemann GmbH & Co KG
3000 Hannover	Com Data
3000 Hannover 1	Trenddata Computer GmbH
3170 Gifhorn	Computer-Haus Gifhorn
3250 Hameln	Witte Bürotechnik
3300 Braunschweig	Computer Studio GmbH
3400 Göttingen	Wiederholt Computer
4000 Düsseldorf	E. Bernshaus GmbH + Co
4000 Düsseldorf	Data Becker
4300 Essen	Helmut Rennen GmbH
4330 Mülheim 12	Compy-Shop
4600 Dortmund	City Elektronik Knupe
4600 Dortmund	Büro Studio Bolz
4800 Bielefeld 1	C S F Computer
5000 Köln	Büromaschinen Braun
5090 Leverkusen	Rolf Rocke Computer
5100 Aachen	Allo Pach GmbH & Co KG
5300 Bonn	Plasman Computercenter
5400 Koblenz	Schmitt Computersysteme
5500 Trier	Bürocenter Lehr GmbH
5900 Siegen	Hees Computer GmbH
6000 Frankfurt	Müller & Nemecek GmbH
6000 Frankfurt	Schmitt Computersysteme GmbH
6100 Darmstadt 13	Heim OHG Bürotechnik
6100 Darmstadt	Schmitt Computersysteme
6200 Wiesbaden	Schmitt Computersysteme
6240 Königstein	KFC Computersysteme
6300 Gießen	Interface Computer
6500 Mainz	Schmitt Computersysteme
6600 Saarbrücken 3	W. N. Pfeiffer EDV Abt.
6700 Ludwigshafen	M K V GmbH
6750 Kaiserslautern	Gotthold Bürocenter
6800 Mannheim	Computer Center
6800 Mannheim	Gauch & Sturm
6900 Heidelberg	Jacom Computer
7000 Stuttgart 31	Kunkel Computer
7022 L.-Echterdingen	Matrai Computer
7100 Heilbronn	Seel's Computerwelt
7100 Heilbronn	Walliser & Co
7150 Backnang	Weeske Computer
7410 Reutlingen	Brock Comuter Shop
7500 Karlsruhe	Papierhaus Erhardt
7530 Pforzheim	DM-Computer GmbH
7600 Offenburg	Frank Leonhardt Electr.
7700 Singen	Udo Meier Computer
7750 Konstanz	Computertechnik Rösler
7890 Waldshut-Tieng.	Hettler DATA
7900 Ulm	Computerstudio Wecker
8000 München 45	Ludwig Computer
8000 München 2	Schulz Computer
8400 Regensburg	C-Soft GmbH
8500 Nürnberg	Technoland
8500 Nürnberg 21	HIB GmbH Computerladen
8700 Würzburg	Schöll Computercenter
8900 Augsburg	Adolf & Schmoll Computer

Computer Technik Kieckbusch GmbH

Am Seeufer 11+22, 5412 Ransbach, Tel. 02623/1618

CH: Hilcu International, Badhausstraße 1
CH-3063 Ittigen/Bern, Tel.: 031/58 66 56

A: Ueberreuter Media, Alserstraße 24, Postfach 306
A-1091 Wien, Tel.: 0222/48 15 38

FAX: 0 26 23/21 40

Floppyspielereien

Teil 5: In den heißen Brei getappt

Willkommen, bienvenue, welcome – kommen Sie, staunen Sie! Erleben Sie die Enthüllung der letzten Rätsel der Floppyprogrammierung! Delektieren Sie sich an den exotischsten Aufzeichnungsverfahren und freuen Sie sich mit mir an den (blauen) Wundern der Technik... treten Sie doch ein....



Bisher habe ich im Floppykurs immer einigermaßen elegant um das im Untertitel erwähnte Nahrungsmittel herum geredet, ohne auf den ganz entscheidenden Punkt zu kommen: Den Floppycontroller. Doch zu meiner Verteidigung muß ich sagen, daß erstens dieses Kapitel wahrscheinlich das komplizierteste ist und deswegen wohl vorbereitet werden mußte; zum Zweiten hat sich einfach im Laufe des Kurses einfach herausgestellt, daß auch die anderen Themen einer breiteren Darstellung bedürfen, damit man sie auch bleibend versteht.

Doch zuguterletzt können wir uns auch diesen Brocken vornehmen. Zuguterletzt, Sie haben richtig gelesen. Ich möchte nach dieser Folge zumindest eine kleine Pause machen, um wieder abseits vom Schreiberischen kreativ zu werden (wer hat da so hämisch gelacht ????). Ob ich die Serie fortsetze, hängt nicht zuletzt von Ihnen ab (schauen Sie sich nicht um, ob jemand anders gemeint sein könnte – ich meine SIE!), genauer: von Ihrer Reaktion und Meinung, telefonisch oder schriftlich bei mir oder bei der ST kundgetan. Zum Zeitpunkt, an dem just diese Zeilen zäh aus meinen Fingern fließen (das war eine krumme Metapher und keine Beschreibung einer anatomischen Absonderlichkeit), habe ich zwar (ohne Schummeln!) durchweg positive Kritiken gehört, aber vielleicht hat sich einfach keiner zu stänkern getraut oder mittlerweile, wenn dieser Teil zu Ihnen gelangt, haben Sie die Nase voll von mir.

Aber wenn ich noch länger herumlabere, kann ich harscher Kritik gewiß sein. Also: Ran an den Speck.

DER KOMMISSAR GEHT UM

Zuerst ein Wort von Kommissar Pingelig, der in den letzten beiden Folgen des Floppykurses ein paar Fehler gefunden hat:

Die Korrektur (betreffend das Listing von HYPERFORMAT) in der ST 7/87, Seite 72, ist selbst korrekturbedürftig. Zum Beispiel muß es in der Routine change_tos "move.b = // xx,BF(a5)" heißen statt "move.b #xx,$6F(a5)". "xx" steht für 21 in der „Alt"-Spalte bzw. 19 in der „Neu"-Spalte. Zudem habe ich eine noch günstigere Lückenbyte-Kombination gefunden, die in den neuesten HYPERFORMAT-Versionen eingebaut ist. Sie lautet 1-2-21-11-2 (und nicht wie im Listing 3-3-21-11-3 bzw. in der Korrektur 1-1-19-9-7). Zudem fehlten in der Korrektur ein paar Zeilen:

```
Seite Alt
1     lea romflag(pc),a2
      cmpi.b #-1,(a2)
      bne rom_sweet_rom
      Neu
      cmpi.l #$00FC0000,$4F2
      beq rom_sweet_rom
```

FLOPPIUS CONTROLLUS – der Schreiberling

Für die neuen Leser fasse ich zusammen, was wir aus den vorherigen Folgen über den Floppycontroller wissen. Vor allem im dritten Teil haben Sie dazu ja schon einiges erfahren. Der Floppycontroller (im folgenden nur noch FDC für Floppy Disk Controller, meine Finger laufen sonst heiß) bekommt über den Umweg des DMA-Controllers Daten vom Prozessor, die er doch bitteschön auf die Diskette schreiben soll. Der Controller kümmert sich selbständig darum, daß die Daten an den richtigen Platz auf einer Spur geschrieben werden (also in den

richtigen Sektor) und daß sie ordentlich kodiert werden. Umgekehrt liefert der Controller beim Lesen dem Prozessor dekodierte Daten von der Diskette. Der FDC ist also der Schreiberling des Systems.

Kodieren? Dekodieren? Könnte man ein gesetztes Bit nicht einfach als Magnetisierung in die eine Richtung und ein Nullbit als anders gerichtete Magnetisierung schreiben? Naja, wenn ich schon so frage...

In Wirklichkeit kommt bei der Aufzeichnung eine weitere Gemeinheit ins Spiel: Die Daten werden kodiert auf die Diskette geschrieben, damit man den Lesevorgang synchronisieren kann.

GEHEIMKODES ENTRÄTSELT!

Wie der Floppycontroller auf der Diskette den Anfang eines Bytes findet, habe ich ja bereits in der letzten Folge zu erläutern versucht. Worum es jetzt geht: Wie finde ich den Anfang eines Bits? Nehmen wir mal an, die Bits würden so auf die Diskette geschrieben wie oben erwähnt. Ein $A1-Byte (dezimal 161, binär 1010 0001) würde dann als Signalpegelverlauf so aussehen (Bild 1):

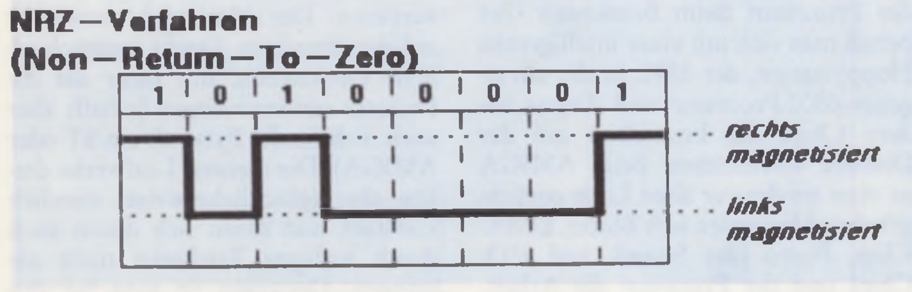


Bild 1: Ein Byte geht auf den Strich

Ich hoffe, das Schema ist auch dem Nicht-Elektroniker klar: Der durchgezogene dicke Strich soll den Verlauf der Aufzeichnung darstellen. Ist der Strich oben (in unserem ersten Beispiel also dann, wenn das zu schreibende Bit gesetzt ist), wird das Stück Plastikscheibe unter dem Lesekopf in die eine Richtung (nennen wir's mal „rechts") magnetisiert. Wenn nicht, wird „links" magnetisiert. Mit Bitzelle bezeichne ich im folgenden den Abschnitt zwischen zwei gestrichelten senkrechten Linien.

Dieses erste Verfahren in Bild 1 heißt NRZ (Non-Return-To-Zero), nicht weil es etwa nie auf den Signalpegel 0 gehen würde, sondern weil es nicht in der Mitte einer Bitzelle auf 0 wechselt, sondern höchstens mal am Rand – zu solchen Kodierungen kommen wir später. NRZ zeichnet sich durch Einfachheit und Unbrauchbarkeit aus. Daß das NRZ einfach ist, sehen Sie schon daran, daß Sie meine weit ausholenden Erläuterungen dazu recht unnötig finden – schließlich liegt das Prinzip auf der Hand. Und unbrauchbar ist es, weil ein Floppycontroller sich an so einem Signal auf der Diskette nicht synchronisieren kann. So sauber wie oben dargestellt sieht eine Aufzeichnung nämlich nie aus – es gibt immer wieder Unregelmäßigkeiten im Material, in der Drehgeschwindigkeit, in der Elektronik des Lesekopfes usw. . Das bewirkt, daß einzelne Abschnitte des obigen Linienzuges gestaucht oder gedehnt werden. Stellen Sie sich vor, das passiert während der vier Null-Bits im Bild 1. Dann läßt sich nicht mehr mit Sicherheit sagen, ob tatsächlich vier oder vielleicht drei oder fünf Null-Bits auf der Diskette stehen (entsprechende Probleme gibt es bei längeren Sequenzen von 1-Bits).

SCHEIBENKLEISTER

Man bräuchte also ein Aufzeichnungsverfahren, bei dem einem der Takt, in dem die Bits auf die Scheibe gekleistert wurden, förmlich in die Äuglein springt. Voil (Bild 2):

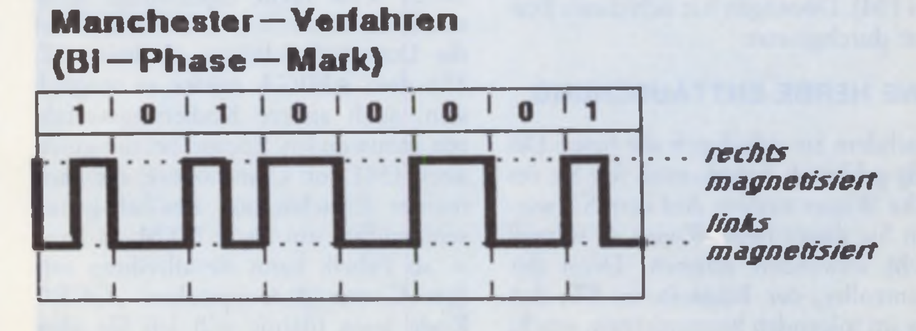


Bild 2: Dasselbe Byte mit etwas mehr Schwung

Beim Manchester- oder Bi-Phase-Mark-Verfahren wird am Anfang einer Bitzelle grundsätzlich der Zustand gewechselt – entweder von „0" (=„links" magnetisiert) auf „1" (=„rechts" magnetisiert) oder umgekehrt, je nachdem, was vorher übertragen wurde. Dadurch erreicht man, daß regelmäßig Pegelwechsel beim Lesen auftreten, an denen sich der Controller orientieren kann. Das eigentliche Datenbit wird in der Mitte der Bitzelle übertragen. Wenn das Datenbit gesetzt ist, wird in der „Bitmitte" ein zusätzlicher Übergang von „0" auf „1" oder umgekehrt erzeugt. Dieses Verfahren ist schon ziemlich zuverlässig, weil keine längeren konstanten Pegel auftreten, denen man keine Information über den Takt der Aufzeichnung (also über die tatsächlichen Längen der einzelnen Bits auf der Diskette) entlocken kann.

Eine ähnliche Technik wurde jahrelang auf den meisten Disketten angewendet: Die FM-Technik (Frequency Modulation). Dabei wird grundsätzlich in der ersten Hälfte der Bitzelle eine „1" erzeugt und in der zweiten Hälfte das eigentliche Datenbit übertragen (Bild 3):

FM- wie Manchesterkode haben den entschiedenen Nachteil, daß sehr viele Übergänge übermittelt werden müssen. Damit man die auch einigermaßen beim Lesen erkennen kann, muß man sich relativ viel Platz auf der Diskette lassen. NRZ- und FM-Kode haben einen weiteren Nachteil gemeinsam: Um einen Datenwert zu erkennen, muß man einen Zustand lesen (beim NRZ-Verfahren eine ganze Bitzelle lang, beim FM-Verfahren in der zweiten Hälfte einer Bitzelle). Wer etwas vom Wesen der magnetischen Induktion weiß, wird mir bestätigen, daß es technisch viel einfacher wäre, wenn der zu übertragende Wert des Bits durch eine Flanke gekennzeichnet wäre, wie beim Manchester-Verfah-

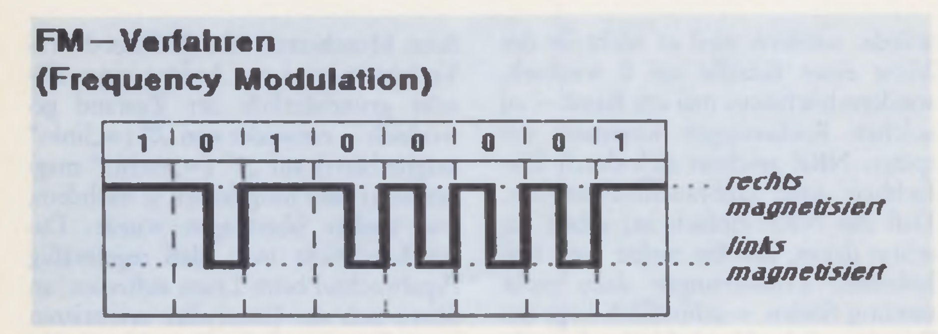


Bild 3: Der klassische Dreh

ren. Wobei anzumerken ist, daß Flanke hier nichts mit Manni Kaltz und schon gar nichts mit Bayern München zu tun hat; Flanke bedeutet hier einen Wechsel von „0" auf „1" oder umgekehrt.

ALLES IN EINEM

Also hat man sich Kodes gesucht, die alles auf einmal bieten: Datensicherheit durch einkodierten Takt, hohe Aufzeichnungsdichte durch seltene Wechsel und einfache Lesbarkeit durch Flankenkodierung. Und man hat sie gefunden (Bild 4).

Wie man sieht (oder auch nicht), ist dieser MFM-Kode (ich vermute: Modified FM-Kode) eher eine Abart des Manchester-Kodes. Wie beim Letzteren wird bei einem gesetzten Datenbit in der Bitmitte der Zustand gewechselt (was ist das also? Richtig: Eine Flanke!); die Flanken am Anfang der Bits (für den Takt) fallen weg – die sind ja auch bei 1-Bits nicht nötig, weil da eh schon eine Flanke anfällt und man somit leicht den Takt erspüren kann. Bei mehreren 0-Bits hintereinander wird die Sache aber kritisch. Deswegen schiebt man ab der zweiten aufeinanderfolgenden Null am Anfang der Bitzelle einen Extraübergang ein.

Bei MFM braucht man nur noch wenig mehr Pegelwechsel als bei NRZ (und damit auch viel weniger Platz als bei FM). Deswegen hat sich dieses Format durchgesetzt.

EINE HERBE ENTTÄUSCHUNG

Nachdem Sie sich durch alle fiesen Details gekämpft haben, muß ich Sie ins kalte Wasser stoßen: Auf dem ST werden Sie dieses neue Wissen so schnell nicht anwenden können. Denn der Controller, der Baustein im ST, den wir im folgenden kennenlernen, macht die MFM-Kodierung ganz alleine – der Programmierer hat darauf keinen Einfluß. Diese Einführung in die Kodierungsverfahren auf magnetischen Datenträgern sollte vielmehr ein Gefühl vermitteln, was der Floppycontroller tatsächlich alles leistet. In den

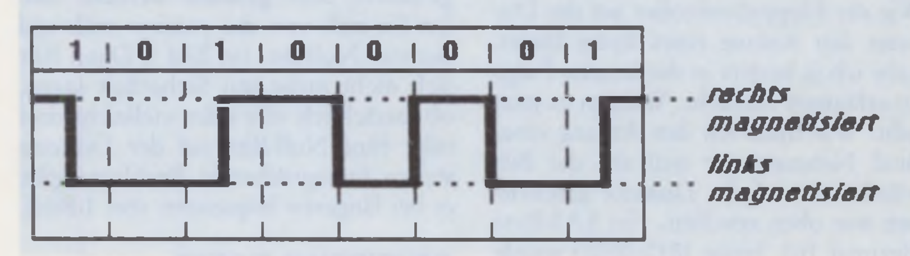


Bild 4: Eine eierlegende Wollmilchsau

Pionierrechnern erledigte diese Arbeit der Prozessor! Beim Brotkasten C64 behalf man sich mit einer intelligenten Floppystation, der 1541, in der ein eigener 6502-Prozessor und diverse andere Chips die Fronarbeit auf der Diskette verrichteten. Beim AMIGA ist man wieder zur alten Linie zurückgekehrt: Hier teilen sich Blitter, DMA-Chip, Portia (der Sound- und I/O-Chip) und der Prozessor die Arbeit, um auch ohne einen „richtigen" Floppycontroller einigermaßen zu Rande zu kommen.

Trotzdem ist das Wissen um den MFM-Kode nicht vergebens. Auf dem AMIGA, zum Beispiel, hat man eben durch seine recht eigenwillige Konstruktion wesentlich mehr Einfluß auf die Datenaufzeichnung als beim ST. Mit dem AMIGA müßte es möglich sein, auch andere Kodierungsverfahren anzuwenden. Ebenso bei der guten alten 1541 von Commodore, die nach meiner Einschätzung flexibel genug sein müßte, um sogar MFM zu lesen – ab Fabrik kann sie allerdings nur den Commodore-typischen „GCR"-Kode lesen (damit will ich Sie aber nicht auch noch quälen).

Ebenso dürfte es wohl einige flexible Controller in der PC-Welt geben. Wer also nebenher noch einen anderen Rechner besitzt, kann sich ja mal kundig machen – und bald gibt es dann überall Diskkonverterprogramme für IBM nach Commodore, AMIGA nach ST und umgekehrt, APPLE-CP/M nach ST-CP/M und so fort – wär das nichts? Damit wäre endlich der kundenfeindliche Spezialitätenfetischismus bei den Computerfirmen überwunden! Und schließlich gibt es ja schon Erweiterungskarten für den ST, an die man die gesamte PC-Peripherie hängen kann. Und da gibt es ja bekanntlich alles, auch flexible Controller...

Nur das Macintosh-Format ist praktisch auf Diskettenebene nicht zu konvertieren. Der Macintosh verwendet auf den einzelnen Tracks unterschiedliche Drehzahlen, um mehr auf die Diskette unterzubringen (schafft aber auch nicht mehr Bytes als ein ST oder AMIGA). Die meisten Laufwerke drehen aber (glücklicherweise) ziemlich konstant und lassen sich davon auch durch heftigste Trickserei nicht abbringen. Deswegen ist man bei den Mac-Emulatoren für den ST auch immer auf die Krücke mit dem RS232-Kabel angewiesen. Aber nun zum Thema des Tages:

IM DIENSTE SEINER MAJESTÄT – DER FLOPPYCONTROLLER

Der WD1772 (so heißt er nämlich) ist eigentlich ein eigener kleiner Prozessor. Deswegen hat er auch interne Register, von denen fünf 8-Bit-Register dem Programmierer zur Verfügung stehen: Statusregister, Trackregister, Sektorregister, Datenregister und Kommandoregister. Track- und Sektorregister sollten intuitiv klar sein. Hier speichert der Controller den aktuellen Track und Sektor, hier kann man ihm aber auch das Ziel einer Operation

mitteilen. Das Datenregister enthält die gerade zu lesenden/schreibenden Bytes sowie bei manchen Befehlen Parameter. In das Kommandoregister kann man nur schreiben, und zwar Befehle (was auch sonst). Das Statusregister kann man nur lesen. Der Status ist wie folgt in die einzelnen Bits kodiert (Tabelle 1):

Bitnr.	Bedeutung
7	MOTOR-ON: 1 = Laufwerksmotor an
6	WRITE-PROTECT: 1 = Schreibschutz drin
5	SPIN-UP: 1 = Motor hat Solldrehzahl (300 Umdrehungen pro Minute) erreicht. Nach *Read-Sector* zeigbt dieses Bit den Typ des Datenfeldes an (0 = normales Datenfeld), 1 = Datenfeld mit „gelöschter" Adreßmarke, siehe unten).
4	RECORD NOT FOUND: 1 = kein korrekter Vorspann gefunden
3	CRC ERROR: 1 = Prüfsummenfehler
2	LOST-DATA: 1 = Datenverluste durch zu lange Wartezeiten Bei Positionierbefehlen bedeutet „gesetzt", daß Track 0 erreicht ist.
1	DATA REQUEST: 1 = Controller meldet, daß er Daten abliefern oder geliefert haben will.
0	BUSY: 1 = Kommando läuft

Tabelle 1: Das Statusregister des Floppycontrollers)

Nun zur Übersicht der Controllerbefehle. Sie werden in das Kommandoregister geschrieben (logo!), dazu braucht der Controller von Fall zu Fall Parameter in anderen Registern.

DIE POSITIONIERBEFEHLE

Sie dienen dazu, den Lesekopf des Laufwerks in die richtige Position zu bringen. Hier eine kurze Zusammenfassung:

RESTORE (in Bits: 0 0 0 0 h V r1 r0): positioniert den Lesekopf auf die Spur 0. Die Buchstaben im Kommandowort bedeuten folgendes:

„h" ist das Motor-On-Flag. Ist es gelöscht, testet der Controller, ob der Motor vielleicht noch vom vorigen Kommando läuft. Wenn nein, fährt er ihn hoch, wenn ja, wird der Befehl gleich ausgeführt. Ist das h-Bit gesetzt, fährt der Controller auf jeden Fall den Motor hoch.

„V" ist das Verify-Flag. Nachdem der Befehl ausgeführt wurde, vergleicht der Controller die Spurnummer im Trackregister mit der Tracknummer, die er im nächsten Vorspannfeld auf der Diskette findet – wenn „V" gesetzt ist. Klappt das Verify nicht, wird das im RECORD NOT FOUND-Bit gemeldet.

„r1" und „r0" geben zusammen an, wie schnell der Controller den Track wechseln soll (abhängig von der Güte des Laufwerks). „00" (also r1=0 und r0=0) bedeutet 2 Millisekunden, „01" (r1=0, r0=1) 3 ms, „10" 5ms und „11" 6ms. Die Laufwerke von ATARI sind für eine Steprate von drei Millisekunden geeignet.

Kurz noch zur Kodierung (für die Nicht-Assembleristen unter Ihnen) ein Beispiel: Das Befehlsbyte 0 bedeutet, daß ein Restore-Befehl ohne jede Raffinessen und mit Steprate 2ms ausgeführt werden soll. 5 (binär 00000101) bedeutet einen Restorebefehl mit Verify und Steprate 3 ms.

SEEK (in Bits: 0 0 0 1 h V r1 r0): Fährt einen bestimmten Track an. Dazu übergibt man vor dem Kommando den gesuchten Track ins Datenregister.

STEP-IN (in Bits: 0 1 0 u h V r1 r0): Bewegt den Lesekopf einen Track nach innen (also zur Mitte der Diskette hin, zu höheren Tracknummern). Das „u"-Bit bewirkt, wenn es gesetzt ist, daß das Spurregister auf dem Laufenden gehalten wird („update-Flag").

STEP-OUT (in Bits: 0 1 1 u h V r1 r0): Raten Sie mal!

STEP (in Bits: 0 0 1 u h V r1 r0): Bewegt den Lesekopf einen Track weit in die Richtung des letzten Step-Befehls.

DIE SEKTORBEFEHLE

READ-SECTOR (1 0 0 m h E 0 0): Vor dem Kommando wird dem Controller die Sektornummer im Sektorregister übergeben. Er liest dann den gewünschten Sektor auf der aktuellen Spur.

Das „E"-Bit gibt dem Lesekopf 30 ms Zeit, um sich auf die Diskette abzusenken. Allerdings liegt der Lesekopf der ST-Laufwerke immer auf, so daß das Bit eigentlich unnötig ist (auf 0 setzen!). Die Entwicklungsdokumentation von ATARI warnt sogar eindringlich davor, dieses Bit zu setzen (warum auch immer).

Das „m"-Bit (m wie multiple) gibt an, ob gleich mehrere Sektoren gelesen (oder beim Write-Sector-Befehl geschrieben) werden sollen. Dazu lädt man die Nummer des ersten Sektors in das Sektorregister, und der Controller liest/schreibt dann alle folgenden Sektoren, bis er keinen mehr findet oder herb unterbrochen wird (dazu später).

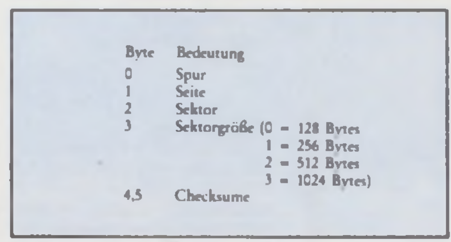
WRITE-SECTOR (1 0 1 m h E P a0): Sektornummer ins Sektorregister, Befehl ins Kommandoregister, und schon rasselt er los: WRITE-SECTOR prüft erst mal nach, ob die Diskette schreibgeschützt ist; wenn ja, läuft gar nichts (wie bei allen Schreibbefehlen). Der Befehl sucht nach dem passenden Sektorvorspann und schreibt dann nacheinander Daten-Adreßmarke, Sektordaten und die Checksumme über die Sektordaten.

„P" gibt an, ob die Daten vorkomprimiert werden sollen (Precompensation-Bit). Ist das Bit = 0, wird komprimiert, wenn nicht, dann nicht. Auf ATARI-Disketten wird immer komprimiert, wegen der höheren Datensicherheit.

„a0" = 0: normale Daten-Adreßmarke schreiben ($FB), a0=1: „gelöschte" Daten-Adreßmarke ($F8) schreiben. Dient zur Unterscheidung von Sektoren. Hier könnte auch ein (primitiver) Kopierschutz ansetzen.

DIE TRACKBEFEHLE

READ-ADDRESS (1 1 0 0 h E 0 0): Liest ein Vorspannfeld ein, das dann so aussieht:



Byte	Bedeutung
0	Spur
1	Seite
2	Sektor
3	Sektorgröße (0 = 128 Bytes 1 = 256 Bytes 2 = 512 Bytes 3 = 1024 Bytes)
4,5	Checksume

Auf der Diskette wird ein Vorspann durch Syncbytes und durch die ID-Adreßmarke ($FE) angekündigt (siehe Floppykurs Teil III). Die Spurnum-

mer wird auch ins Sektorregister geladen.

READ-TRACK (1 1 1 0 h E 0 0): Liest eine ganze Spur ein. Dieser Befehl ist aber sehr unzuverlässig. Verwenden kann man ihn eigentlich nur, um Abstände zwischen Daten auf der Diskette zu messen. Die Daten selbst sind mit Vorsicht zu genießen. Beim Lesen einer ganzen Spur erkennt der Controller nämlich viel zu oft imaginäre Synchronisationsbytes (er ist halt ein Sensibelchen) und verfälscht darauf folgende Bytes. Erst beim nächsten tatsächlich vorhandenen Syncbyte erkennt er den Fehler. Nach dem zweiten gelesenen real existierenden Synchronisationsbyte hat sich der FDC gefangen und liest ein paar korrekte Bytes. Das bedeutet, daß mindestens das dritte Synchronisationsbytes richtig gelesen wird. Daran kann man sich zumindest orientieren.

WRITE-TRACK (1 1 1 1 h E P 0): Eine Spur schreiben, also formatieren. Man übergibt dem Controller einen ganzen Track voll Daten und Bruchteile von Sekunden hat man etwas völlig anderes auf der Spur stehen. Das liegt daran, daß alle Bytes von $F5 bis $FF als Steuerbytes interpretiert werden. Was sie bewirken? Das hier (Tabelle 2):

Byte	Bedeutung/Aktion
$F5	Syncbyte $A1 schreiben, Prüfsummenbildung neu starten
$F6	Syncbyte $C2 schreiben
$F7	Prüfsumme schreiben (2 Bytes)
$F8	gelöschte Daten-Adreßmarke (jetzt kommen Daten!)
$F9	Marke (keine spezielle Bedeutung ersichtlich)
$FA	Marke (ebenso gesichtslos)
$FB	normale Datenadreßmarke (jetzt kommen Daten!)
$FC	Marke (tja, auch nix Besonderes)
$FD	Marke (siehe oben)
$FE	ID-Adreßmarke (jetzt kommt ein Vorspann!)
$FF	Ende des Datenblocks

Tabelle 2: Steuerbytes beim Write-Track-Befehl)

DRUM PRÜFE, WER SICH EWIG SCHINDET...

Ein paar Takte zur Prüfsummenbildung: Sie geschieht automatisch im Controller selbst. Wenn man dem Controller beim Schreiben einer Spur ein Syncbyte $A1 übergibt, löscht er intern die Checksumme und fängt neu mit ihrer Berechnung an. Jedes jetzt ankommende Byte wird durch eine Hardwaremühle gedreht (das Verfahren nennt sich CRC, Cyclic Redundancy Check), die eine Prüfsumme erzeugt. Sendet man ein $F7-Byte, wird diese Checksumme 16 Bit breit auf die Diskette geschrieben (für diesen Zweck hat der FDC übrigens ein internes Register, das 16 Bit breit ist). Auch der Write-Sector-Befehl erstellt eine solche Prüfsumme. Beim Lesen funktioniert das Ganze dann entsprechend anders herum.

EIN EIGENTÜMLICHER GESELLE

Ein Sonderling unter den Controllerbefehlen ist...

FORCE INTERRUPT (1 1 0 1 I3 I2 I1 I0): Löst eine Unterbrechung des aktuellen Controllerbefehls aus. I0 und I1 tun nichts zur Sache, sie müssen 0 sein. Wenn I2 gesetzt ist, wird bei jeder Umdrehung ein Interrupt ausgelöst (damit kann man die Umdrehungsgeschwindigkeit messen). Wenn I3 gesetzt ist, wird das laufende Kommando mit einer Interruptanforderung des Controllers beendet. Sind I2 und I3 gleich 0, wird das laufende Kommando ohne Interrupt abgewürgt. Puh, das war's. Sie sind noch dabei? Dann darf ich Ihnen zu Ihrem Durchhaltevermögen gratulieren. Wie Sie die Befehle anwenden, können Sie aus dem Listing LOCKSLEY.S in der vorigen Folge des Floppykurses erfahren. Für diejenigen, die diesen legendären Artikel nicht gelesen haben, ein kurzes Anwendungsbeispiel (Bild 5).

```
* Floppyroutine für den Step-Befehl
* (Beispiel)
* muß im Supervisormodus laufen, der Floppy-VBL muß abgewürgt
* werden (z. B. mit st $43e)

daccess = $FFFF8604
dmodus  = $FFFF8606
mfp     = $FFFFFA01

move.w #%00110001,d7    * Step-Befehl mit Track-Update, Steprate 3 ms
move.w #$180,dmodus     * Kommandoregister auswählen
                        * (siehe Teil III des Floppykurses)
bsr wrfdc               * d7 an DMA bzw. FDC schicken

move.l #$40000,d7       * Timeout-Konstante
schau_nach:
btst #5,mfp             * Fertigmeldung des FDC am MFP68901?
beq fertig              * Jawohl, Kommando ist ausgeführt
subq.l #1,d7            * Timeoutzähler dekrementieren
bne schau_nach          * Noch nicht auf 0, dann weiter warten

move.b #$d0,d7          * FORCE-IRQ-Befehl
bsr wrfdc               * d7 an DMA bzw. FDC schicken
move.w #250,d1          * Zähler auf 250
bsr wait                * IRQ-Befehl Zeit lassen

fertig:
rts                     * und Schluß

move.w #30,d1           * Zähler auf 30
bsr wait                * Kurze Warteschleife (30 Durchläufe)
move.w d7,daccess       * Befehl an DMA bzw. FDC schicken
move.w #30,d1           * Zähler auf 30
bsr wait                * Warteschleife (wie oben)
rts                     * und zurück
```

Bild 5: Steptanz mit dem Lesekopf

Diese Routine ist nicht ganz bis zum bitteren Ende ausprogrammiert, das Drumherum (vor allem die Zeitschleife, das Einschalten des Supervisormodus et cetera) finden Sie vollständig im Listing zu LOCKSLEY.S (Teil III des Floppykurses).

Zu Beginn wird das DMA-Modus-Register so eingestellt, daß das Kommandoregister des FDC offen vor uns im DACCESS-Register vor uns liegt. Dort plazieren wir den STEP-Befehl selbst (mit bsr wrfdc). Die WRFDC-Routine beinhaltet auch noch zwei vorsichtige Warteschleifen (besser vorbeugen als ärgern) vor und nach dem Beschreiben des DACCESS-Registers, vor allem, um dem FDC Zeit zu lassen, seine Fertigmeldung vom vorangegangenen Befehl zurückzuziehen. Im folgenden fragen wir nämlich in einer Schleife genau diese Fertigmeldung (im Bit 5 des ersten Registers des MFP-68901) ab. Fragen wir zu fix nach dem Kommando auf 'Fertig' ab, hat der FDC eventuell noch gar nicht realisiert, daß da ein Kommando angekommen ist, und tut so, als wäre er schon fertig. Die Warteschleife wird abgebrochen, wenn die Fertigmeldung innerhalb einer bestimmten Zeit ankommt, die durch die Timeout-Konstante $40000 gegeben ist. Ansonsten wird das laufende Kommando brutal abgewürgt (mit einem FORCE-IRQ-Befehl) und die Routine verlassen.

NOCH EIN KOPIERSCHUTZ

Mit dem EXTENDED MINIMON und LOCKSLEY.S aus den vorherigen Folgen können Sie (in den Grenzen des Controllers) völlig beliebig formatierte Disketten erzeugen. Zum Beispiel können Sie im Vorspannfeld jedes Sektors für die Größe des nachfolgenden Sektors nicht nur die Zahlen 0–3 verwenden, sondern auch höhere. Allerdings werden von den höheren Zahlen auch wieder nur die untersten beiden Bits beachtet, das heißt, eine 4 entspricht einer 0 (also einer Sektorgröße von 128 Bytes). Das kann man für einen Kopierschutz verwenden:

1. Man schreibt statt einer 2 für die Sektorgröße (512 Bytes) eine 6 in das Sektorgrößenfeld des Vorspanns (Headers).

2. Beim Programmstart liest man mit Read-Address den Vorspann des betreffenden Sektors ein und prüft, ob dort eine 6 steht.

3. Ein unachtsames Kopierprogramm erkennt möglicherweise nur, daß der Sektor 512 Bytes lang ist und setzt drum leichtsinnigerweise eine 2 ins Sektorgrößenfeld. Eine Kopie kann man deswegen leicht erkennen.

POLIERARBEITEN

Schließlich bin ich Ihnen noch eine Ergänzung des EXTENDED MINIMON aus der letzten Folge schuldig. Mit dieser Erweiterung bekommen Sie die Möglichkeit, von BASIC aus sogar die einzelnen Register von Floppycontroller und DMA-Chip zu lesen und zu modifizieren. Sie finden die dazu notwendigen MINIMON-Erweiterungen in den Listings 1 und 2. Im Quellkode von SELECT.S, der Selektierroutine des MINIMON, hatte ich dafür mit Bedacht eine Abfrage auf erweiterte Steuerkodes vorgesehen. Sie müssen also nur die Routine extend ein bißchen abändern und die neu hinzugekommenen Routinen eintippen. Für BASIC-Fans gibt es in Listing 2 eine Datawüste des gesamten SELECT.S, mit denen Sie die alten SELECT-Datazeilen im EXTENDED MINIMON ersetzen können. Zusätzlich ist im Listing 2 auch eine kleine BASIC-Routine (für den EXTENDED MINIMON) abgedruckt, mit der Sie die einzelnen Register lesen und beschreiben können.

Wie steuert man diese Erweiterung an? Normalerweise übergibt man der SELECT-Routine in der Variablen Laufwerk (eigentlich ist das eine relativ feste Speicherstelle, im Basicprogramm sel+3) Zahlen von 0 bis 7, die Seite und Nummer des Laufwerks angeben, das angewählt werden soll. Übergibt man Zahlen, die größer als 7 sind, spricht die Abfrage auf „extended commands" an. „8" bedeutet, daß Register des Floppycontrollers gelesen werden sollen, „9", daß Register geschrieben werden. Mit „10" kann man das DMA-Statusregister lesen.

Bei den Leseroutinen bekommt man generell den Inhalt des gewünschten Registers in temp (sel+8 für die BASIC-Fans) zurück. Das gewünschte Register übergibt man in parm (sel+4). Bei den Schreibroutinen steht vor dem Aufruf der zu schreibende Wert in temp. Beispiel in BASIC (Bild 6):

Sie möchten wissen, welches Register welche Nummer hat? Ganz einfach (Tabelle 3):

Register	Nummer
Kommandoregister (bzw. FDC-Statusregister)	128 (= $80)
Trackregister	130 (= $82)
FDC-Sektorregister	132 (= $84)
Datenregister	134 (= $86)
DMA-Sektorregister	144 (= $90)

Tabelle 3: Die Register-Nummern des Floppycontrollers)

Diese Tabelle entspricht derjenigen, die ich bereits in der vorherigen Folge für die Registernummern angegeben habe.

DEM KOPIERSCHUTZ AN DIE GURGEL

Damit sind Sie mit einem recht kompletten Werkzeug zur Analyse Ihrer Disketten ausgerüstet. Eigene Erweiterungen des EXTENDED MINIMON sind leicht anzufügen. Interessant wäre eine Analyseroutine, die kopiergeschützte Disketten auf Tricks abklopft. Eine Anregung, wie das zu machen wäre:

Wie Sie ja jetzt wissen, ist der READ-TRACK-Befehl des FDC ganz und gar nicht zuverlässig. Schade, denn wenn er perfekt funktionieren würde, könnten wir das Problem mit einem einzigen READ-TRACK erschlagen. Leider aber muß man sich behelfen. Immerhin hat man den READ-ADDRESS-Befehl, der Auskunft über die vorhandenen Sektorvorspänne gibt. Hat man erst mal die Nummern der Sektoren, kann man versuchen, diese Sektoren mit READ-SECTOR einzulesen. Schließlich ist es oft noch wichtig zu wissen, wie groß die Lücken zwischen den Sektoren sowie zwischen Vorspann und Sektor sind. Das kann man nun ungefähr ermitteln, indem man die Spur zusätzlich mit READ-TRACK

```
POKE SEL+3,9        ! EXTENDED-Modus für 'Register schreiben'
DPOKE SEL+4,128     ! Kommandoregister selektieren
DPOKE SEL+8,33      ! Zu schreibender Wert (Step-Befehl)
CALL SEL
```

Bild 6: Alle Register gezogen

einliest und die Abstände zwischen den Syncbytes mißt (spätestens das dritte Syncbyte liest der Controller nämlich korrekt ein, auch bei READ-TRACK!).

Trotzdem ist beim WD1772, dem Controller des ST, oft viel Hirnschmalz nötig, bis man aus den ulkigen, beinahe zufälligen Mustern, die der READ-TRACK-Befehl liest, die Kopierschutztricks rekonstruiert hat. Individuelle Genialität hat hier also noch Chancen, ein Grund, warum die Knackerei auf dem ST noch nicht ausgestorben ist. Am Besten wäre es natürlich, wenn kein Kopierschutz mehr nötig wäre, aber bis das erreicht ist, werden uns die Softwarehersteller wohl noch vor so manche Denksportaufgabe stellen. Manche Leute, die es wissen müssen, meinen, daß man beim Knacken am Schnellsten seinen Rechner kennenlernt.

Wie gesagt, dies ist vorläufig die letzte Folge des Floppykurses. Für die angekündigten Vorhaben wie das Kopierprogramm für HYPER-FORMAT-Disketten, das superschnelle Formatierprogramm und andere Gags brauche ich zunächst ein bißchen Freiraum. Was von meinen Projekten Realität wird, können Sie in unregelmäßigen Abständen in der ST lesen. Keep hacking!

Claus Brod

```
' Erweiterung für den EXTENDED MINIMON
' aus der ST 10/87
' zusätzliche Registerroutinen
' erweitertes SELECT.S in den Datazeilen
'
' Written 1987 by Claus Brod
'                 Am Felsenkeller 2
'                 8772 Marktheidenfeld
'                 09391/3206
'
' Written in GfA-Basic
'
Procedure Register
  ' sel+3:8 für lesen, 9 für schreiben, 10 für DMA-Status
  ' in sel+4 wird das Register übergeben
  ' in sel+8 wird der Wert übergeben
  Print R$;"Register";O$
  Print R$;"L";O$;"esen oder ";R$;"S";O$;"chreiben?"
  Repeat
    A$=Upper$(Input$(1))
  Until A$="L" Or A$="S"
  If A$="L"
    @Get_them_all
  Else
    Op%=9
    Poke Sel+3,Op%
    Print "(1) Kommandoregister"
    Print "(2) Trackregister"
    Print "(3) FDC-Sektorregister"
    Print "(4) Datenregister"
    Print "(5) DMA-Sektorregister"
    Repeat
      A=Val(Input$(1))
    Until A>0 And A<6
    R%=(A-1)*2
    If A=5
      R%=16
    Endif
    Dpoke Sel+4,128+R%
    If Op%=9
      Input "Neuer Wert für das Register";W%
      Dpoke Sel+8,W%
    Endif
    Call Sel
    Print "Opcode:   $";Hex$(Peek(Sel+3))
    Print "Register: $";Hex$(Dpeek(Sel+4))
    Print "Wert:     $";Hex$(Dpeek(Sel+8) And 255)
    Print
  Endif
Return
'
Procedure Get_them_all
  Poke Sel+3,8                                        ! Register lesen
  Dpoke Sel+4,128+0
  Call Sel
  Print "FDC-Statusregister: $";Hex$(Dpeek(Sel+8) And 255)
  Dpoke Sel+4,128+2
  Call Sel
  Print "Trackregister:      $";Hex$(Dpeek(Sel+8) And 255)
  Dpoke Sel+4,128+4
  Call Sel
  Print "FDC-Sektorregister: $";Hex$(Dpeek(Sel+8) And 255)
  Dpoke Sel+4,128+6
  Call Sel
  Print "Datenregister:      $";Hex$(Dpeek(Sel+8) And 255)
  Dpoke Sel+4,128+16
  Call Sel
  Print "DMA-Sektorregister: $";Hex$(Dpeek(Sel+8) And 255)
  Poke Sel+3,10                                       ! DMA-Status lesen
  Call Sel
  Print "DMA-Status:         $";Hex$(Dpeek(Sel+8) And 255)
Return
'
'
Select:
Data 60,E,0,0,0,0,0,0,0,0,0,0,0,0,0,0
Data 48,E7,FF,FE,42,80,61,6A,45,FA,FF,F2,24,80,50,F9
Data 0,0,4,3E,3E,3A,FF,DC,66,14,33,FC,0,80,0,FF
Data 86,6,32,39,0,FF,86,4,8,1,0,7,66,F4,C,7
Data 0,8,6C,5C,A,7,0,7,CE,3C,0,7,40,E7,0,7C
Data 7,0,13,FC,0,E,0,FF,88,0,10,39,0,FF,88,0
Data C0,3C,0,F8,8E,0,13,C7,0,FF,88,2,51,F9,0,0
Data 4,3E,46,DF,45,FA,FF,96,20,12,61,6,4C,DF,7F,FF
Data 4E,75,2F,0,3F,3C,0,20,4E,41,5C,8F,4E,75,51,C9
Data FF,FE,4E,75,2F,A,3F,3C,0,9,4E,41,5C,8F,4E,75
Data C,7,0,8,67,14,C,7,0,9,67,20,C,7,0,A
Data 67,52,45,FA,0,60,61,DC,60,BA,33,FA,FF,48,0,FF
Data 86,6,61,1C,45,FA,FF,42,34,80,60,A8,3E,3A,FF,3A
Data CE,7C,0,FF,33,FA,FF,2E,0,FF,86,6,61,14,60,94
Data 32,3C,0,1E,61,A8,30,39,0,FF,86,4,32,3C,0,1E
Data 60,9C,32,3C,0,1E,61,96,33,C7,0,FF,86,4,32,3C
Data 0,1E,60,8A,33,FC,0,90,0,FF,86,6,30,39,0,FF
Data 86,6,45,FA,FE,F4,34,80,60,0,FF,5A,52,6F,75,74
Data 69,6E,65,20,6E,6F,63,68,20,6E,69,63,68,74,20,69
Data 6D,70,6C,65,6D,65,6E,74,69,65,72,74,2E,D,A,0
Data 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
Data ***
'
```

NEUES FUTTER FÜR DEN EXTENDED MINIMON

Listing 1 beinhaltet die Routinen, die den EXTENDED MINIMON (vorgestellt in der ST 7/87 und ST 9/87) um die REGISTER-Funktion erweitern. Laden Sie Ihren EXTENDED MINIMON und tippen Sie einfach die neuen Routinen dazu. Die alten SELECT-Datazeilen löschen Sie bitte und ersetzen sie durch die im Listing 1 angegebenen. Ebenso müssen Sie die Prozedur Register durch die neue ersetzen.

Mit 'R' können Sie im Menü des Trackmonitors die neuen Routinen ansteuern. Sie können zwischen Lesen ('L') und Schreiben ('S') auswählen. Nach 'L' erhalten Sie die Inhalte aller Register des Floppycontrollers. Nach 'S' können Sie zunächst auswählen, welches Register Sie neu beschreiben wollen, bevor Sie den Wert eingeben. Nach der Eingabe bekommen Sie eine Rückmeldung.

Die Schreibroutine ist eigentlich nur der Vollständigkeit halber angefügt. Mit der Leseroutine können Sie aber beispielsweise sehr schön beobachten, welche Bits des Statusregisters der FDC nach einer Weile zurücksetzt, wenn er nichts zu tun hat. Im Menü des Trackmonitors bekommt man zwar auch eine Rückmeldung über den Status des FDC, allerdings handelt es sich hier um den Status während der Ausführung des Kommandos.

```
****************************
* Erweiterung für SELECT.S
* Register des FDC und DMA-Chips
* lesen und schreiben
*
* Written 1987 by Claus Brod
*                 Am Felsenkeller 2
*                 8772 Marktheidenfeld
*                 09391/3206
*
* Assembliert mit AS68
****************************
```

```
****************************
* extend: Erhält Kommando in d7
*
****************************
extend:
    cmpi.b #8,d7                 * Register lesen?
    beq rdreg                    * Ja, zur Registerleseroutine
    cmpi.b #9,d7                 * Register schreiben?
    beq wrreg                    * Ja, zur Registerschreiberoutine
    cmpi.b #10,d7                * DMA-Status lesen?
    beq dmastat                  * ja, zur DMA-Statusroutine
```

```
    lea nochnit(pc),a2              * Routine noch nicht
    bsr msg                         * implementiert
    bra raushier                    * und raus

***************************
* in parm angegebenes Register lesen
* und in temp ablegen
***************************
rdreg:
    move.w parm(pc),dmodus          * Register auswählen
    bsr rdfdc                       * FDC/DMA-Register lesen
    lea temp(pc),a2                 * Adresse des Ausgabefeldes
    move.w d0,(a2)                  * Registerinhalt schreiben
    bra raushier

***************************
* in parm angegebenes Register mit dem
* Wert aus temp beschreiben
***************************
wrreg:
    move.w temp(pc),d7              * Wert holen
    and.w #$ff,d7                   * Nur unteres Byte beachten
    move.w parm(pc),dmodus          * Register auswählen
    bsr wrfdc                       * d7 zum FDC schicken
    bra raushier                    * und raus
```

```
***************************
* rdfdc: FDC-Register lesen
***************************
rdfdc:
    move.w #30,d1                   * 30 Schleifen
    bsr time                        * warten
    move.w daccess,d0               * Register lesen
    move.w #30,d1                   * 30mal rundherum
    bra time                        * um Ulm und in Ulm und
                                               um Ulm herum
***************************
* wrfdc: FDC-Register beschreiben
***************************
wrfdc:
    move.w #30,d1                   * 30 Loops
    bsr time                        * warten
    move.w d7,daccess               * d7 ins Register
    move.w #30,d1                   * 30 Loops
    bra time                        * warten und raus

***************************
* dmastat: DMA-Status lesen
***************************
dmastat:
    move.w #$90,dmodus              * DMA-Sektorregister anwählen
    move.w dmodus,d0                * DMA-Status lesen
    lea temp(pc),a2                 * Adresse des Ausgabefeldes
    move.w d0,(a2)                  * Status merken
    bra raushier                    * und raus
```

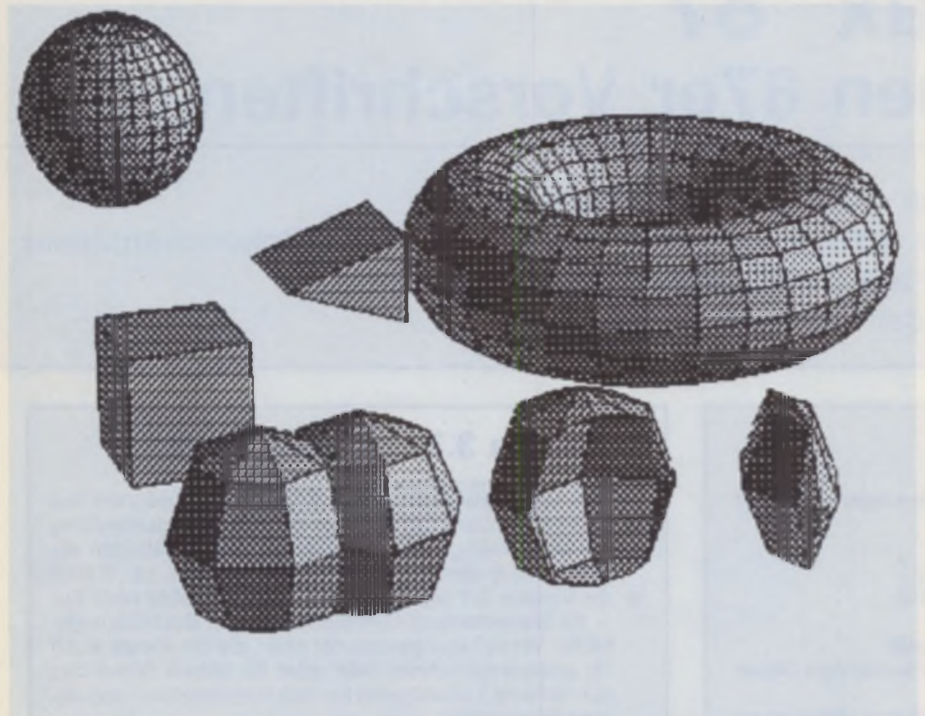
MEHR ALS NUR SELEKTIEREN

Wie schon im Artikel beschrieben, nutze ich für die Erweiterung der SELECT-Routine aus der ST9/87 die 'open ends' aus, die ich absichtlich in dieser Routine belassen hatte.

Laden Sie die alte SELECT-Routine in Ihren Editor, löschen Sie die alte EXTEND-Routine und ersetzen Sie sie durch die neue. Die übrigen Routinen werden einfach nur hinzugefügt. Danach kann man mit den fiktiven Laufwerksnummern 8,9 und 10 die zusätzlichen Funktionen aufrufen (siehe Haupttext).

Das Konzept der fiktiven Laufwerksnummern ist natürlich beliebig erweiterbar, Sie können in der EXTEND-Routine ohne Schwierigkeiten zum Beispiel die imaginäre Drivenummer 11 verwenden, um eine eigene Routine anzuspringen, sei es eine Analyseroutine für einen Track oder ein kleines Kopierprogramm oder was auch immer. Lassen Sie sich was einfallen!

Bilderspiele:
Faszination Computergrafik



Der zweite Teil unserer Grafikserie hat sich wegen einer ausgedehnten Urlaubsreise des Autors leider etwas verzögert. Nachdem jetzt aber die gesamte Urlaubsfaulheit abgeschüttelt wurde, geht es weiter (und richtig los).

In der letzten Folge hatten wir uns mit dem Unterschied zwischen raster- und objektorientierten Grafiksystemen beschäftigt. Um diese Unterschiede noch einmal kurz zusammenzufassen: Ein rasterorientiertes System weiß nicht was es tut, es dient lediglich dazu, Werkzeuge, die direkt ein Raster, wie zum Beispiel den Bildschirm, mit Werten füllen, zur Verfügung zu stellen. Diese Methode hat den Nachteil, daß sie Informationen über die Beschaffenheit des Rasters voraussetzt und außerdem nicht ohne weiteres Änderungen an Einzelteilen der Darstellung erlaubt, da es ja nicht weiß, was überhaupt dargestellt ist.

Im Gegensatz dazu basiert ein objektorientiertes System darauf, daß vom Anwender die Objekte, die dargestellt werden sollen, beschrieben werden. Die Darstellung des Objektes, das heißt, die Umsetzung in das für das jeweilige Ausgabegerät geeignete Raster- oder Vektorformat übernimmt dann das System. Um ein einzelnes Objekt der Zeichnung zu verändern, genügt es, seine Beschreibung zu editieren, das System kann dann eine neue Zeichnung anfertigen.

Daraus folgt, daß auch für ein objektorientiertes System Zeichenroutinen erforderlich sind, die ein Objekt auf dem Bildschirm darstellen. Um diesen Kern herum müssen aber Routinen liegen, die 1. die Objekte verwalten, 2. die Transformierung in ein für die Malroutinen geeignetes Format erledigen.

Im Betriebssystem des ST sind eine ganze Reihe von Zeichenroutinen bereits vorhanden, mit deren Hilfe es eigentlich nicht sehr schwierig ist, ein rasterorientiertes Grafikprogramm zu schreiben. Das beweist auch die Flut von Malprogrammen für den ST.

Wie sehen aber nun grafische Objekte aus, wie kann man sie darstellen und manipulieren?

Das einfachste aller Objekte ist mit Sicherheit der Punkt. Um einen Punkt zu beschreiben, verwendet man ein Koordinatensystem (Bild 1). Die Koordinaten eines Punktes beschreiben seine Lage in bezug auf einen Ausgangspunkt, den Koordinatenursprung. Kompliziertere Objekte beschreibt man im einfachsten Fall als eine Menge von miteinander durch Linien verbundenen Punkten (Bild 2).

In den Bildern 3–6 sehen Sie einige Möglichkeiten, mit denen man Punkte

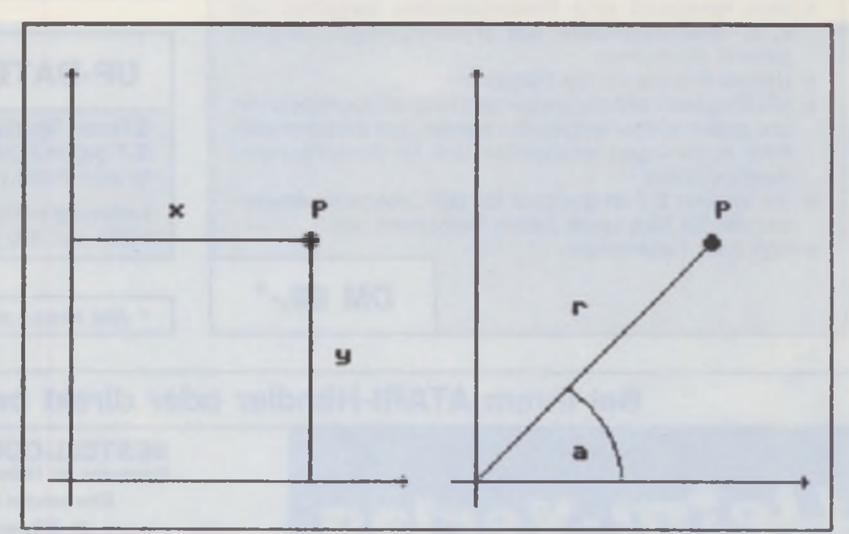


Bild 1)
Zwei Koordinatensysteme.
Links das sogenannte kartesische Koordinatensystem, in dem die Lage eines Punktes durch den Abstand auf der x- und y-Achse vom Ursprung aus gemessen beschrieben wird.
Rechts das Polarkoordinatensystem. Die Lage des Punktes wird durch die Länge der Strecke r, die vom Ursprung zum Punkt P reicht, sowie den Winkel a zwischen x-Achse und r bestimmt.

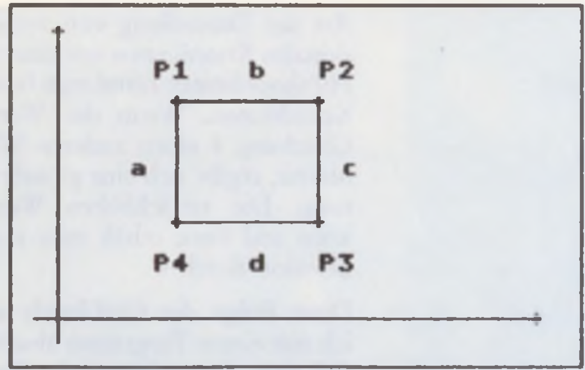


Bild 2:
Das Objekt Viereck läßt sich mit den vier Punkten P1 bis P4, die jeweils aus einer x- und y-Koordinate bestehen, sowie einer Liste von Linien, die jeweils aus einem Anfangs und Endpunkt bestehen, beschreiben.

oder Objekte aus Punkten und Linien im Koordinatensystem manipulieren kann. Es läßt sich leicht nachweisen, daß es völlig genügt, die Endpunkte einer Linie zu manipulieren, um damit die ganze Linie analog mitzuverändern. Es ist also nicht nötig, alle Punkte einer Linie zu bearbeiten. In einer realen Welt wäre es auch unmöglich, besteht doch eine Linie aus unendlich vielen Punkten.

Wie man in den Bildern sieht, gibt es eine ganze Menge von Möglichkeiten, einen Punkt zu manipulieren. Es wäre natürlich schön, wenn es für alle diese Möglichkeiten eine geeignete mathematische Methode gäbe, die es erlaubte, alle Operationen 1. einheitlich und 2. auf einmal zu erledigen. Glücklicherweise gibt es tatsächlich eine solche Möglichkeit. Bitte haben Sie Verständnis dafür, daß die folgenden Erklärungen aus Platzmangel nicht dafür gedacht sind, das Rechnen mit Vektoren und Matrizen lückenlos zu erklären. Es gibt eine Menge Mathebücher, die sozusagen sehnsüchtig auf diese Gelegenheit gewartet haben. Im folgenden soll nur kurz die Praxis erläutert werden.

Fassen wir doch einmal die Gleichungen aus den obigen Bildern zusammen. Für die Bilder 3–5 ergibt sich:

(1) xneu = a★x + b★y + m und
(2) yneu = c★x + d★y + n

Dabei sind die Variablen a und d für eine Skalierung, b und c für eine Verzerrung und m und n für eine Verschiebung verantwortlich. Es ist leicht ersichtlich, daß eine Rotation dann entsteht, wenn in der ersten Gleichung

a = cos(phi) und b = −sin(phi) ist und in der Zweiten c = sin(phi) und d = −cos(phi).

Offensichtlich ist also eine Rotation um den Ursprung bereits eine kombinierte Operation, die aus einer Skalierung und einer Verzerrung besteht.

Die beiden Gleichungen (1) und (2) bilden zusammen eine Einheit, die man ein lineares Gleichungssystem nennt. Ein lineares Gleichungssystem kann man aber auch anders formulieren:

(3) P2 = T★P1 mit P1 = (x,y),P2 = (xneu,yneu), $T = \begin{matrix} a & b \\ c & d \end{matrix}$

In dieser Schreibweise sind P2 und P1 Vektoren und T eine Matrix, die die nötigen Parameter enthält. Eigentlich ist dies nichts weiter als eine Kompaktschreibweise für ein lineares Gleichungssystem. Ihnen wird sicherlich auffallen, daß die Parameter für die Translation in der Matrix T fehlen. In einer Matrix aus 2★2 Elementen gibt es leider keine Möglichkeit, Translationen einzubauen. Dieses Problem ist aber leicht lösbar, doch dazu später mehr.

Zur Erklärung der Multiplikation einer Matrix mit einem Vektor betrach-

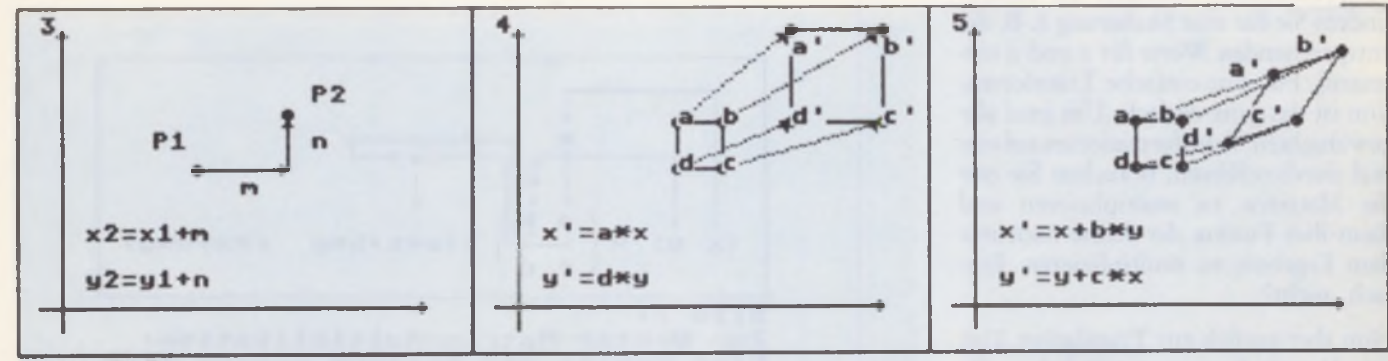


Bild 3-5:
Einfache Koordinatentransformationen.
3: Translation: Ein Punkt wird verschoben durch Addition von m zur x-Koordinate und/oder n zur y-Koordinate.
4: Skalierung: Eine Skalierung ergibt sich durch Multiplikation von x und y mit einem festen Faktor. Die Multiplikation ergibt gleichzeitig eine Verschiebung und eine Skalierung, weil Punkte mit großer x-Koordinate weiter verschoben werden, als Punkte mit kleiner x-Koordinate. Entsprechendes gilt für die y-Koordinate.
5: Verzerrung: Der Verzerrungseffekt entsteht durch die Addition eines Summanden, der von der einen Koordinate verändert wird, zur anderen Koordinaten. Dabei bleiben parallele Seiten parallel, aus einem Quadrat wird ein Parallelogramm.

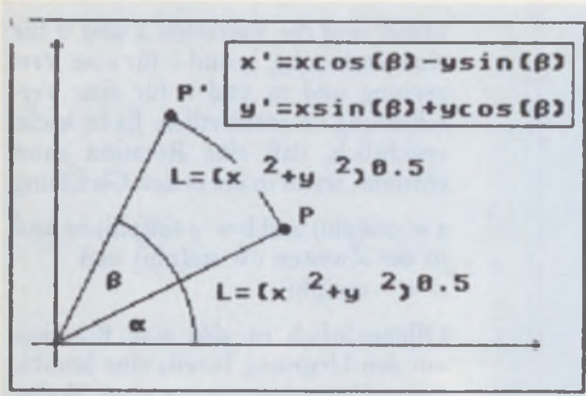


Bild 6:
Rotation: Die Berechnung der Matrixkoeffizienten für eine Rotation ist etwas aufwendiger Es gilt:
sin(α) = y / L und (1)
cos(α) = x / L (2)
Wenn der Punkt P nun um den Winkel β rotiert werden soll, ist der Gesamtwinkel von der x-Achse aus gesehen α+β. Also gilt:
y'/L=sin(α+β) und (3)
x'/L=cos(α+β). (4)
Aus den Sätzen über die Winkelfunktionen wissen wir:
sin(α+β)=cos(β)*sin(α)+sin(α)*cos(β) (5)
und
cos(α+β)=cos(β)*cos(α)+sin(β)*sin(α). (6)
Setzt man die Gleichungen (1) und (2) ein, ergibt sich:
y'/L=cos(β)*y/L+sin(β)*x/L und
x'/L=cos(β)*x/L-sin(β)*y/L.
Wenn man nun mit L multipliziert, ergeben sich die Gleichungen aus obiger Zeichnung.

ten Sie bitte Bild 7. Das wirklich praktische an einer Matrix ist aber, daß man sie nicht nur mit Vektoren, sondern auch mit anderen Matrizen multiplizieren kann (Bild 8). Wollen Sie also verschiedene Transformationen kombinieren, brauchen Sie nur die entsprechenden Matrizen zu basteln (indem Sie für eine Skalierung z. B. die entsprechenden Werte für a und d einsetzen). Für eine einfache Transformation ist das ganz einfach. Um jetzt alle gewünschten Transformationen auf einmal durchzuführen, brauchen Sie nur die Matrizen zu multiplizieren und dann ihre Punkte der Reihe nach mit dem Ergebnis zu multiplizieren. Einfach, nicht?

Nun aber zurück zur Translation. Um eine Verschiebung zu ermöglichen, fügen wir dem Punktvektor P einfach eine dritte Komponente, die den Wert h haben soll, und der Matrix T eine dritte Reihe und Spalte von Parametern zu:

$$(4)\ (xneu,yneu,h) = \begin{matrix} a & b & m \\ c & d & n \\ 0 & 0 & s \end{matrix} \star (x,y,1), s=1$$

Ausmultipliziert ergibt das:

xneu = a★x + b★y + m
yneu = c★x + d★y + n
h = 0 + 0 + 1

Da die dritte Zeile sich nicht verändert hat, können wir sie ignorieren. Diese Art der Darstellung von zweidimensionalen Koordinaten mit einer dritten Hilfskoordinaten nennt man homogene Koordinaten. Wenn der Wert s in Gleichung 4 einen anderen Wert annimmt, ergibt sich eine globale Skalierung: Die tatsächlichen Werte für xneu und yneu erhält man erst nach Division durch s.

Diese Folge der Grafikserie möchte ich mit einem Programm abschließen, daß zum Ausprobieren der Transformationsmatrix dient. Es ist in Pascal geschrieben, verwendet aber keine Funktionen, die man nicht in GfA- oder Omikron-Basic simulieren könnte. Auch die Übersetzung nach C dürfte kein Problem sein.

Das Programm benutzt eine Technik, die in großen Grafiksystemen üblich ist, weil die Ausgabe der Grafiken auf verschiedenen Geräten möglich sein muß: Die Speicherung und Transformierung des Beispielobjektes erfolgt in einem sogenannten normalisierten Koordinatensystem, das so auf dem Bildschirm nicht darstellbar ist. Erst zum Zeitpunkt der Ausgabe werden die Bildpunkte in das Bildschirmkoordinatensystem übersetzt. Damit ist das Programm extrem leicht an verschiedene Ausgabegeräte und -auflösungen anpaßbar. Das verwendete System reicht auf beiden Koordinatenachsen von 0 bis 1. Einzelne Punktkoordinaten müssen also dazwischen liegen.

Das Programm enthält einige allgemeine Routinen zur Bearbeitung von Matrizen:

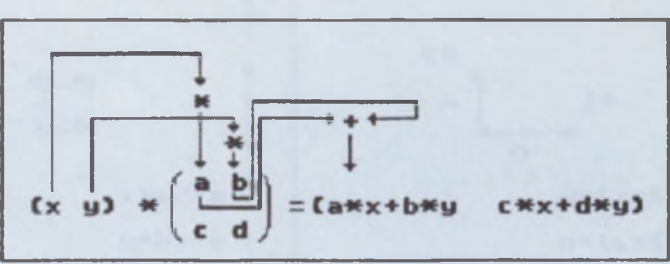


Bild 7:
Zur Vektor-Matrix-Multiplikation: Eine Spalte des Vektors ergibt sich nach folgender Formel:

$$v(i)= \sum_{j=1}^{n} m(i,j)*u(j)$$

Dabei ist u der Ursprungs- und v der Ergebnisvektor. m ist eine Matrix mit n Spalten. i ist der Index für die Vektorspalte. Jede Vektorspalte v(i) ergibt sich aus der Summe der Produkte der Elemente der i-ten Matrixzeile und des Ursprungsvektors.

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} * \begin{pmatrix} e & f \\ g & h \end{pmatrix} = \begin{pmatrix} a*e + b*g & usw. \\ usw. & usw. \end{pmatrix}$$

Bild 8:
Zur Matrixmultiplikation:
Es gilt folgende Formel:

$$e(i,j) = \sum_{k=1}^{n} a(i,k)*b(k,j)$$

Dabei ist e die Ergebnismatrix, a und b die zu multiplizierenden Matrizen. Jedes Matrixelement e(i,j) ergibt sich aus n Summanden. Das Element e(1,2) der Matrix aus obigem Beispiel ist also:

e(1,2) = a*f + b*h.

Eine Umsetzung dieser Formel in einen Algorithmus finden Sie im Beispielprogramm.

Die Prozedur null_mat2dh füllt eine aus 3★3 Elementen (Die Abkürzung 2dh am Ende ihres Namens bedeutet '2-Dimensional, homogen' – das wird sich noch ändern, wenn der Raum dazukommt!) bestehende Matrix mit Nullen. Eine Nullmatrix verhält sich wie eine Null beim normalen Rechnen: Alles was damit multipliziert wird, ergibt Null.

ident_mat2dh macht aus einer Matrix eine Einheitsmatrix. Das ist eine Matrix, die sich wie die 1 bei der normalen Multiplikation verhält. Frage: Wie muß eine solche Matrix aussehen? (Folgen Sie den Rechenregeln!)

null_vec2dh – das gleiche für einen Vektor, was null_mat2dh für eine Matrix ist.

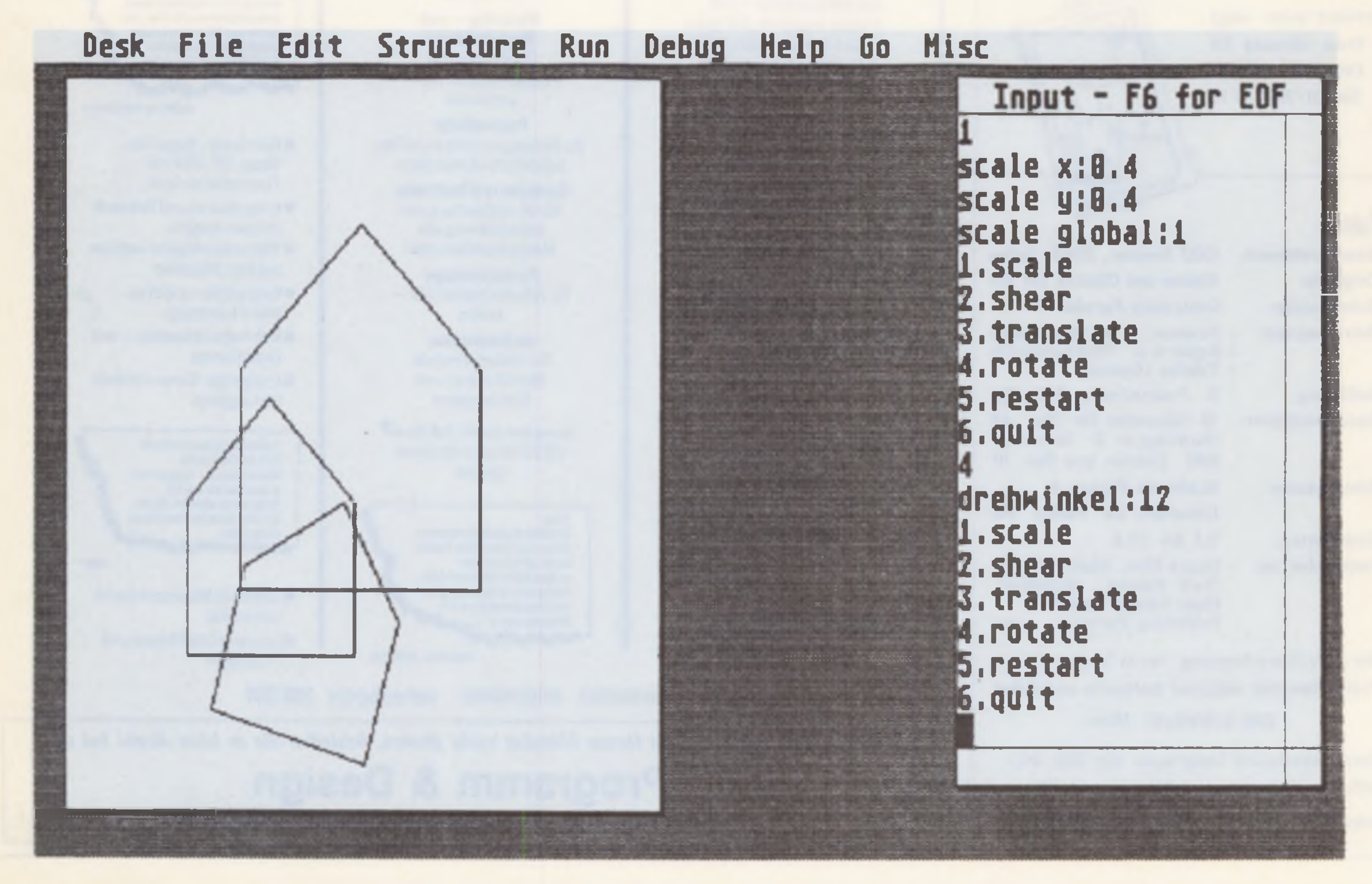
matmul2dh: Multipliziert zwei Matrizen.

vecmat2dh: Multipliziert einen Vektor mit einer Matrix.

Wie funktioniert dieses primitive Grafiksystem? Zuerst einmal wird, wie immer, alles initialisiert. Dabei wird die Transformationsmatrix t mit den Werten der Einheitsmatrix aufgefüllt.

Dann wird das Originalbild gezeichnet. In einer Schleife kann jetzt die Matrix t mit den wildesten Tansformationen gefüllt und das Ergebnis angezeigt werden. Dabei wird immer wieder das Original transformiert, nicht eine Transformation nach der anderen mit dem bereits transformierten Bild ausgeführt. Mit der 'restart'-Option kann die Matrix t wieder initialisiert werden.

Der Ablauf der Transformationen ist der folgende: Zuerst füllt jede der Routinen set_scale, set_trans, set_rot und set_shear eine eigene Matrix mit den eingegebenen Werten. Diese Matrix wird dann mit t multipliziert und das Ergebnis in t zurückgeschrieben. Damit enthält t nach dem Aufruf einer der set_ Prozeduren eine neue Kombination von Transformationen.



Dann wird die Prozedur do__transform aufgerufen. Sie multipliziert jeden der Punkte des Objektes mit t. Danach wird noch die Korrektur für den globalen Scaling-Faktor s durchgeführt. Das Ergebnis ist das manipulierte Objekt, das sich allerdings immer noch im normalisierten Koordinatensystem befindet.

Deshalb wird die Prozedur do__view aufgerufen. Von hier aus wird die view__trans-Funktion aufgerufen, die die normalisierten Koordinaten in Bildschirmkoordinaten übersetzt. Am Anfang des Programmes befinden sich die Konstanten, die die dafür notwendigen Werte des Bildschirmkoordinatensystems enthalten. Danach wird das Objekt gezeichnet.

Dieses Verfahren der aufeinanderfolgenden Transformationen ist wichtig. Um das Programm an 3D-Transformationen anzupassen, muß nur eine Transformation eingefügt werden, die die dreidimensionalen Koordinaten auf eine Fläche projeziert. Natürlich müssen auch die Datentypen geändert werden. Aber der prinzipielle Aufbau bleibt gleich. Hier sieht man auch den großen Vorteil von Matrizen-Mathematik noch einmal sehr deutlich: Für die Anpassung an drei Dimensionen muß ich nur die Größe der Matrizen und der Vektoren ändern. Die Routinen sehen ansonsten völlig gleich aus.

Für eine Anpassung an andere Auflösungen müssen also nur die Parameter für eine Prozedur geändert werden. Eventuell muß auch die Formel in dieser Prozedur geändert werden, aber nur dann, wenn sich das Koordinatensytem des Ausgabegerätes erheblich von dem des Bildschirmes unterscheidet.

Drei weitere Prozeduren sind systemabhängig: Im Hauptteil des Programmes wird eine Routine zum Löschen des Bildschirms aufgerufen. Die do__view-Routine zeichnet Linien mit Hilfe einer Funktion draw. Schließlich wird in der Prozedur init die Grafikausgabe des verwendeten Pascal-Systems initialisiert. Dabei handelt es sich um Alice, einen Pascal-Interpreter, den ich zum Testen von Algorithmen nur empfehlen kann.

Im folgenden die Änderungen für CCD-Pascal, die ich aus Zeitmangel allerdings nicht mehr testen konnte (Weitere Hinweise in den Kommentaren im Programm):

1. Koordinaten-Konstanten: xmax:639; ymax:399; xstart:0; ystart:399

2. ClrScr(Fenster) ersetzen durch Clear__Screen

3. Draw(x1,y1,x2,y2) ersetzen durch line(x1,y1,x2,y2,1,0,0,0,1mode,0).

'lmode' muß vorher als $FFFF definiert werden.

Das Programm als 'TOS' compilieren. Das wär's.

Zum allerletzten Schluß noch ein paar Aufgaben:

1. Wie sieht die Matrix für eine Spiegelung eines Punktes an der x- oder y-Achse des Koordinatensystems aus?

2. Welchen geometrischen Sinn könnten die Parameter der Matrix aus Gleichung 4 haben, die dort den Wert 0 besitzen?

3. Schreiben Sie eine Routine, die ein Objekt um seinen Mittelpunkt und nicht um den Nullpunkt rotieren läßt. Lösen Sie das Problem allgemein, also: Wie kann man ein Objekt um einen beliebigen Punkt rotieren lassen?

4. Ändern Sie das normalisierte Koordinatensystem des Beispielprogramms so, daß die Werte von −1 bis +1 reichen und der Nullpunkt in der Bildschirmmitte liegt. Schauen Sie sich dazu die Formeln in der view__trans-Prozedur an und ändern Sie sie.

In der nächsten Folge werden wir uns mit der dritten Dimension und möglichen Strukturen von Objekten befassen. Es wäre ja langweilig, immer nur das gleiche Haus zu verwenden...

```
program transform(input, output);

|Programm zum Test von Transformationen mit homogenen Koordinaten|
|------------------------------------------------------------------
---------------|
|Das Programm verwendet fuer alle Berechnungen ein normalisiertes |
|Koordinatensystem, das in beiden Richtungen von 0 bis 1 reicht|
|Vor der Ausgabe werden die Bildpunkte dann in Bildschirmkoordinaten|
|uebersetzt. Zur Anpassung an andere Grafikaufloesungen muss daher |
|fast nichts geaendert werden. |
|------------------------------------------------------------------
---------------|
|Dieses Programm wurde mit ALICE geschrieben und für CCD-Pascal ange-
|paßt (Aus Zeitmangel: Nicht schön, aber es läuft).|
| Achtung: Kein Clipping !!! Die Anpassung der |
|Grafikroutinen an eine andere Ausgabeeinheit ist  extrem einfach|
|1. Setzen Sie die Aufloesungswerte ihres Bildschirms oder Grafikfenster|
|bei xmax und ymax ein. Den Abstand der linken unteren Ecke vom
absoluten |
|Bildschirmursprung setzen Sie bei xstart und ystart ein. Achtung: die in|
|der Routine View_trans verwendete Formel geht davon aus, dass das
System|
|Ihres Bildschirms seinen Ursprung in der linken oberen Ecke hat, wie auf|
|dem ST ueblich.Ist dies nicht der Fall, muss die Formel geaendert werden|
|2. Die Routine do_view benutzt eine procedure line, die Linien von x1,y1|
|zu x2,y2 zieht. Der Aufruf ist der komplette Line-A-Aufruf, die meisten|
|anderen Sprachen bieten einfachere Aufrufe (z.B: Draw(x1,y1,x2,y2) |
|Setzen Sie einfach die entsprechende Funktion ihrer Programier-|
|sprache ein.|
|3. Im Hauptprogramm und an anderen Stellen wird eine Escape-Sequenz |
| verwendet, um den Bildschirm zu löschen.|
|Auch hier die entsprechende Routine fuer Ihre Sprache einsetzen.|
|------------------------------------------------------------------
---------------|
|Auch eine Umsetzung in GfA- oder Omikron-Basic ist kein Problem;|
|Es wurden keine Pascal-Spezialitaeten benutzt.|
|------------------------------------------------------------------
---------------|
```

```
|ALICE verlangt die Initialisierung aller Variablen vor der ersten|
|Benutzung. In Basic oder anderen Pascal-Dialekten kann auf einige der|
|Initialisierungen (z.B. die Routine init_bild) verzichtet werden|
|------------------------------------------------------------------
---------------|

const
        xmax = 639;  |Diese Zahlen sind die Aufloesung des monochromen|
        ymax = 399:  |Bildschirms. Hier einfach die Werte|
        xstart = 0;  |fuer das Zielfenster einsetzen.|
        ystart = 399;

type
        mat2dh = array [1..3] of array [1..3] of real;
        vec2dh = array [1..3] of real;
        data = array [1..5] of vec2dh;

var
        original : data;
        bild : data;
        t : mat2dh;   |t ist die Transformationsmatrix|
        fenster : text;
        g : char;
        ende : Boolean;
        l_style: integer; |Für CCD-Pascal|

procedure init ;
        |Diese Routine oeffnet ein Grafikfenster und leitet alle Grafik-|
        |ausgaben dorthin um. Dieses Fenster simuliert dabei den ganzen|
        |Atari-Schirm in mittlerer Aufloesung. In der CCD-Version sind|
        |diese Befehle auskommentiert. Der Write-Befehl löscht den|
        |Bildschirm mit einer Escape-Sequenz|
        |und l_style erhält das linienmuster für die line-a-Grafik.|
begin
       | SetCoordinate(4);|
       |Koordinatensystem 640*200. Ursprung oben links|
       |page ;|
       |loescht den Textbildschirm|
```

```
        |QuickWindow(fenster, 0, 3);|
        |malt eine weisse Box und meldet sie als Fenster an|
        |GraphicsWindow(fenster);|
        |simuliert den gesamten Grafikbildschirm in dem soeben geoeffneten|
        |Fenster - Aufloesung 640*200 Punkte, wie oben festgelegt|
        write(chr(27),'E');
        l_style:=$FFFF;
end;

procedure init_original ;
        |Hier wird ein Bild initialisiert|
begin
        original|1||1| := 0.3;
        original|1||2| := 0.3;
        original|1||3| := 1;
        original|2||1| := 0.3;
        original|2||2| := 0.6;
        original|2||3| := 1;
        original|3||1| := 0.5;
        original|3||2| := 0.8;
        original|3||3| := 1;
        original|4||1| := 0.7;
        original|4||2| := 0.6;
        original|4||3| := 1;
        original|5||1| := 0.7;
        original|5||2| := 0.3;
        original|5||3| := 1;
end;

procedure init_bild ;
        |Initialisierung-s.o.|
begin
        bild|1||1| := 1;
        bild|1||2| := 1;
        bild|1||3| := 1;
        bild|2||1| := 1;
        bild|2||2| := 1;
        bild|2||3| := 1;
        bild|3||1| := 1;
        bild|3||2| := 1;
        bild|3||3| := 1;
        bild|4||1| := 1;
        bild|4||2| := 1;
        bild|4||3| := 1;
        bild|5||1| := 1;
        bild|5||2| := 1;
        bild|5||3| := 1;
end;

procedure ident_mat(var matrix: mat2dh);
        |Diese Routine erzeugt eine Einheitsmatrix|
        var
                i, j : integer;

begin
        for i := 1 to 3 do begin
                for j := 1 to 3 do begin
                        if i = j then begin
                                matrix|i||j| := 1;

                        end
                         else begin
                                matrix|i||j| := 0;
                        end;
                end;
        end;
end;

procedure null_mat(var matrix: mat2dh);
 |Erzeugt eine Nullmatrix-ist wahrscheinlich nur in ALICE erforderlich|
        var
                i, j : integer;

begin
        for i := 1 to 3 do begin
                for j := 1 to 3 do begin


                        matrix|i||j| := 0;
                end;
        end;
end;

procedure null_vec(var v: vec2dh);
        |Auch hier: Alice verlangt Initialisierung: Nullvektor|
        var
                i : integer;

begin
        for i := 1 to 3 do begin
                v|i| := 0;
        end;
end;

procedure matmul2dh(m1, m2: mat2dh; var m3: mat2dh);
        |Multipliziert die 3*3 Matrizen m1 und m2; Ergebnis in m3|
        var
                i, j, k : integer;

begin
        for i := 1 to 3 do begin
                for k := 1 to 3 do begin
                        for j := 1 to 3 do begin
                                m3|i||k| := m3|i||k| + m1|i||j|*m2|j||k|;
                        end;
                end;
        end;
end;

procedure vecmat2dh(v: vec2dh; m: mat2dh; var w: vec2dh);
        |Multipliziert den Vektor v mit der Matrix m, Ergebnis in w|
        var
                j : integer;

begin
        for j := 1 to 3 do begin
                w|j| := v|1|*m|1||j| + v|2|*m|2||j| + v|3|*m|3||j|;
        end;
end;

procedure do_transform(m: mat2dh);
        |Multipliziert die Punkte des Originals mit der Transformations-|
        |matrix t und verwandelt die homogenen in normale Koordinaten|
        var
                i, j : integer;
 begin
        for i := 1 to 5 do begin
                vecmat2dh(original|i|, m, bild|i|);
                for j := 1 to 2 do begin
                        bild|i||j| := bild|i||j|/bild|i||3|;
                end;
        end;
 end;

 procedure view_trans(var v: vec2dh; var x1, y1: integer);
  |Rechnet die normalisierten Koordinaten (x und y von 0 bis 1) in|
  |Bildschirmkoordinaten um|
  |--------------------------------------------------------------
---------|
  |Diese Routine muss zur Anpassung an andere Grafik geaendert werden|
  |wenn der Ursprung nicht in der linken oberen Ecke liegt|
 begin
        x1 := round(v|1|*xmax + xstart);
     |Hier werden die Weltkoordinaten in Bildschirmkoordinaten uebersetzt|
        y1 := -round(v|2|*ymax - ystart);
    |Die Formel fuer y ist so fuer den Nullpunkt in der oberen linken Ecke|
    |gedacht. In normalen Systemen analog zur x-Formel setzen|
 end;

 procedure do_view;
  |Gibt das von do_transform erzeugte Bild auf dem Schirm aus. Benutzt |
  |view_trans zur Uebersetzung der normalisierten Koordinaten|
        var
                i, k : integer;
                x1, x2, y1, y2 : integer;

 begin
        for i := 1 to 5 do begin
                if i = 5 then begin
                        k := -4;
                end
                 else begin
                        k := 1;
                end;
                view_trans(bild|i|, x1, y1);
                view_trans(bild|i + k|, x2, y2);
                line(x1,y1,x2,y2,1,0,0,0,l_style,1);
        |line zieht eine Linie von x1,y1 zu x2,y2 und muss zur Anpassung|
        |durch eine entsprechende Routine ersetzt werden - Hier ist es |
        |ein direktes line-a-Binding, deswegen die zusätzlichen Parameter|
        end;
 end;

 procedure set_scale ;


        |Eingabe eines Skalierungswertes; dann Berechnung von t|
        var
                x, y, s : real;
                m : mat2dh;

 begin
        ident_mat(m);
        write('scale x:');
        readln( x);
        m|1||1| := m|1||1|*x;
        write('scale y:');
        readln( y);
        m|2||2| := m|2||2|*y;
        write('scale global:');
        readln( s);
        m|3||3| := m|3||3|*s;
        matmul2dh(t, m, t);
 end;

 procedure set_shear ;
        |Eingabe von Verzerrungswerten und Berechnung von t|
        var
                m : mat2dh;
                x, y : real;

 begin
        ident_mat(m);
        write('shear x:');
        readln( x);
```

```
        m[1][2] := m[1][2] + x;
        write('shear y:');
        readln( y);
        m[2][1] := m[2][1] + y;
        matmul2dh(t, m, t);
end.

procedure set_trans ;
        {Verschiebungseingabe und Berechnung von t}
        var
                m : mat2dh;
                x, y : real;

begin
        ident_mat(m);
        write('trans. x:');
        readln( x);
        m[3][1] := m[3][1] + x;
        write('trans.y:');
        readln( y);
        m[3][2] := m[3][2] + y;
        matmul2dh(t, m, t);
end;

procedure set_rot ;
        {Und schliesslich Drehungen; Berechnung von t}
        var
                m : mat2dh;
                a : real;
                c, s : real;
                n : real;
                i : integer;

begin
        ident_mat(m);
        write('drehwinkel:');
        readln( a);
        c := cos(a);
        s := sin(a);
        for i := 1 to 3 do begin
                n := m[i][1]*c - m[i][2]*s;
                m[i][2] := m[i][1]*s + m[i][2]*c;
                m[i][1] := n;
        end;
        matmul2dh(t, m, t);
end;

begin
     ende := false;
     ident_mat(t);
     init ;
     init_original ;
     init_bild ;
     {Anzeige des Originalbildes}
     do_transform(t);
     do_view ;
{In dieser Schleife koennen verschiedene Transformationen ausprobiert}
{werden. Kombinationen sind moeglich - erst 'restart' loescht t}
     repeat
        write(chr(27),'H'); {Escape-Sequenz für CCD-Pascal-Cursor Home}
        writeln('1.scale');
        writeln('2.shear');
        writeln('3.translate');
        writeln('4.rotate');
        writeln('5.restart');
        writeln('6.quit');
        readln(g);
        case g of
             '1': begin
                     set_scale ;
             end;
             '2': begin
                     set_shear ;
             end;
             '3': begin
                     set_trans ;
             end;
             '4': begin
                     set_rot ;
             end;
             '5': begin
                    ident_mat(t);
                    Write(chr(27),'E');
          { loescht den Bildschirm - hier also entsprechende }
           {Routine aufrufen}
            end;
            '6': begin
                    ende := true;
            end;
        end;
        do_transform(t);
        do_view ;
    until ende = true;
end.
```

Top ST-Software von Herbysoft

ST Video, der Hit:
Eingabe bis zu 5000 Titeln, lauffähig in Farbe und SW. Sortierung nach 3 Kriterien, Haupt-, Leih- und Rückgabedatei, Listendruck. Infofile über: Anzahl der Filme und Cassetten (diff. nach 180 240 etc) Restlaufzeiten aller Cassetten (wahlweise mit Ausdruck).
Bestell-Nr. HL 3000 / Preis 69 DM

ST-Mathemat:
Prozentrechnung, Zinsrechnung, Grundrechnen, Berechnung von Flächen und Körpern (wahlweise mit Ausdruck)
Bestell-Nr. HL 1000 / Preis 49 DM

ST-Mathetrainer:
Das ideale Lernprogramm für Schüler der Klassen 1 – 5. Einmaleins, Grundrechnen sowie (über Dateien) Umrechnung von Gewichten und Längenmaßen. Die Dateien können selbst editiert werden. Komplette Kontrolle über wahlweisen Ausdruck, Endbenotung, Zeitvorgabe.
Bestell-Nr. HL 2000 – Preis 49 DM

ST-Rechtschreiben:
Drei Dateien zur Übung des Rechtschreibens, übrige Merkmale wie bei ST Mathetrainer. Zusätzlich Schwerpunktdatei in der falsch beantwortete Fragen gespeichert werden.
Bestell-Nr. HL 4000 / Preis 49 DM

ST-Benzinverbrauch:
Programm zur Erfassung der Autokosten. Benzin, Öl und Reparaturkosten können getrennt für 2 Fahrzeuge erfaßt werden. Berechnung von Verbrauch pro 100 Km. Listendruck, Einzeldruck.
Bestell-Nr. HL 5000 / Preis 39 DM

Alle Programme sind in hoher und mittlerer Auflösung lauffähig und können sehr komfortabel und einfach über Menüleiste bedient werden. Bestellung bei:

HERBERT LAMMERS
Hubertusstr. 6 · 6200 Wiesbaden 12 · Tel. 0 61 21 / 6 25 35
INFO GEGEN FREIUMSCHLAG

PROFESSIONELLE SOFTWARE FÜR ATARI ST

TKC TERMIN/ADRESS verwaltet alle Adressen und Termine. Alle Felder sind frei benennbar, einfaches Umfunktionieren in z. B. eine LP-Verwaltung. Serienbriefe mit 1st Word, Suchen mit Wildcards, voll unter GEM. Bedienung über Maus oder Tastatur.
Sensationeller Preis **DM 149,-**

TKC-HAUSHALT verwaltet alle Einnahmen und Ausgaben in bis zu 80 Konten unter GEM. Ausgabe von Kontenblättern, Monats- u. Jahresbilanzen.
Incl. Handbuch kostet TKC-Haushalt **DM 129,-**

EINNAHMEN/ÜBERSCHUSS ST erleichtert die Buchführung, erstellt Bilanzen und Ausdrucke. 3 MwSt-Sätze voreinstellbar. Eingabe netto oder brutto, incl. Handbuch.
Preis: **DM 149,-**

TKC-VIDEO Verwaltet Videofilme unter GEM. Ausgabemasken können selbst erstellt werden. Lieferung incl. Handbuch.
Preis: **DM 79,-**

ST-KEYMASTER Tastaturumbelegung, incl. Desktopuhr und Druckeranpassung **DM 49,-**

ST-VOKABELTRAINER Spezielle Tastaturbelegung z. B. für FRANZÖSISCH! **DM 49,-**

BEI IHREM ATARI-HÄNDLER ODER DIREKT GEGEN VORKASSE (FREI) ODER NACHNAHME (ZZGL. DM 5,–)

TK Computer-Technik Thomas Kaschadt
Bischofsheimer Str. 17 · 6097 Trebur-Astheim
Telefon: 0 61 47 / 550

GEHEIM-Tips für den Atari ST

copySTar v2.2 *Neue Version*

- Kopiert **alle** Atari ST Programme, **normale** und **kopiergeschützte**
- **Superschnelle Kopien** von "normalen" Disketten unter 30 Sekunden!!
- **Superschnelle Formatierung**, SS 16 sec
- Konvertiert in **Spezialformat** für **doppelte Geschwindigkeit** ohne zusätzliche Hardware
- Optionale Erweiterung der Diskettenkapazität um 50/100 oder 90/180 Kilobyte (SS/DS)
- Testmöglichkeit der Drehzahl des Laufwerkes
- Überprüfung von Disketten auf defekte Bereiche (Qualitätstest)
- Neue Kopierschutzerkennung für die neuesten Schutzverfahren!
- Kinderleichte Bedienung durch GEM
- Update-Service! Jeder Kunde wird automatisch benachrichtigt
- Ausführliches Handbuch in Deutsch – kein Kopierschutz!

nur DM 169,–

T.L.D.U. v1.3 *Neue Version*
The Last Disc Utility

- Endlich eine **komplette Disk-Utility** mit allen Funktionen
- Von vielen Fachzeitschriften getestet und für SEHR GUT befunden
- Natürlich auch mit Hard- und Ramdisk
- Voll programmierbar ('C'-ähnliche Makrosprache mit Editor)
- Echte (Sub)-directory und Fat-Funktionen
- 10 Buffer, Disassembler, Folgemodus, Suchen, Vergleichen, Drucken, Umfangreiche Makro-Bibliothek und mehr
- Ein- und Ausgabe in ASCII, EBCDIC, HEX oder DEZIMAL-FORMAT
- Deutsches Handbuch – kein Kopierschutz

nur DM 149,–

Detective v1.0

- Erstellt **Assemblersource-Dateien** von jedem Maschinenprogramm
- Verwaltet Text-, Data-, BSS-Bereiche, **Symbole**, Marker und mehr
- GEM-Bedienung – deutsches Handbuch – kein Kopierschutz
- Ein Muß für jeden ernsthaften Programmierer

nur DM 149,–

Wo? Natürlich bei Ihrem Atari-Händler! oder direkt bei STARSOFT Hannover

Versand in die ganze Welt!



Im Gosewinkel 20 · D-3000 Hannover 61
☎ 05 11 / 56 25 26

ATARI ST: Die Massenspeicher kommen...

20 MB Festplatte anschlußfertig .... DM 1.398,–
20 MB Streamer anschlußfertig .... DM 1.698,–
20 MB Festplatte + 20 MB Streamer ..... DM 2.998,–
anschlußfertiges Subsystem!

P M D **0 81 06 / 3 39 41**
Matthias Aures · Postfach 10 01 05 · 8011 Vaterstetten

Verlangen Sie mehr!

LOGO auf dem Atari ST

Dieter und Jürgen Geiß
Logo auf dem Atari ST
1986, 145 S., kart., DM 35,–
ISBN 3-7785-1262-5
Dieses Buch beweist, daß das Atari-LOGO weit mehr als eine Kindersprache ist. Hier stehen die Antworten auf Fragen, die im Original-Handbuch offen geblieben sind.



Dieter und Jürgen Geiß
Software-entwicklung auf dem Atari ST Programmieren unter GEM und TOS
2. Aufl. 1987, 422 S., kart., DM 54,–
ISBN 3-7785-1533-0
Dieses Buch enthält alles, was ein ernsthafter Programmierer braucht, um große Software auf dem Atari ST zu entwickeln. Nach dem Durcharbeiten dieses Buches werden Sie in der Lage sein, professionelle Software mit allen Finessen auf dem Atari ST zu erstellen.
NEU: jetzt mit Omikron-BASIC

Hajo Lemcke, Volker Dittmar und Michael Sommer
Programmierlexikon für den Atari ST
1986, ca. 450 S., kart. DM 48,–
ISBN 3-7785-1412-1
Das Lexikon enthält zu jedem Stichwort nicht nur wie üblich eine Beschreibung, sondern auch eine Programmieranleitung. Viele Tabellen und Querverweise erleichtern die Arbeit. Sie finden alles über GEM, VDI, AES, Chips, Schnittstellen, BIOS, XBIOS, GEMDOS, Systemvariablen und die Line-A Graphikbefehle.

BESTELLCOUPON
einsenden an: Dr. Alfred Hüthig Verlag, Im Weiher 10, 6900 Heidelberg

Titel ____
Name, Vorname ____
Straße, Nr. ____
PLZ, Ort ____
Datum, Unterschrift ____



Kleinanzeigen

BIETE HARDWARE

ROM-TOS 70,– DM, 05141/214161

SF 314 neu 450,– DM 02225/892332

Die ATARI-MAILBOX:
0571/710141

SF 354 DM 150,–
Tel. 07582/8127 ab 18 Uhr

Verk. SF 314 v. PC-Gehäuse 350,– DM
Tel.: 069/391636 ab 17 Uhr

---

520 STM ROMTOS, V7, 354, NEC 1036, SM 124, EPSON LX 86, CDI-HITRANS, Bücher, Softw., 50 Disk, 30 Zeitschr., sonstiges, 1A-Zustand, 7 Mon. alt nur komplett, NP: 4000,– DM, für FP: 2300,– DM, 5090 Leverkusen, Tel.: 0214/9257

EPROMMER mit Textool 145,– DM
ACCU-UHR nur 35,– DM ☎ 040/862771

ATARI 520 ST + 1 Megabyte, TOS im ROM, inkl. Floppy, Maus u. Monitor-Adapterkabel, n. 800,– DM, mit Garantie, Tel.: 06742/1887

SF 354 – 150,– DM
TV-Modulator – DM 150,–
J. Koppitsch, Tel.: 0241/504458

DRUCKER-SCANNER, ☎ 0711/744775

SF 314 VB 450,–, ☎ 02331/72081

Verk. ATARI 260 ST – Kompl. System
M. Michels 02642/23511 >19.00 Uhr

★HF-MODULATOR★ ☎ 02301/4374

ATARI 1040 ST ROM-TOS, SM 124, Harddisk SH 204 bootbar, PROLOG LOGO, 1st Word plus, The PAWN, Winter Games + 5 MB PD + Util. für nur DM 3300,– 06421/47657

MEGA 4 ST, NEU! 3300,– DM
Tel.: 06721/42112

PANASONIC KX-P 1091, VB 300,– DM wenig benutzt. Tel. 08152/70619

ST-Komplettverkauf: 0202/595154

★ NEC-1036 Diskstationen ★
★ Einzellaufwerk 375,– ★
★ Doppellaufwerk 675,– ★
★ eingeb.Netzt.,superfl. ★
★ Betr.Kontroll. grau ★
★ Softw. billig – Liste! ★
★ ATARIPOINT 07031/52607 ★

Diskettenzugriffe bis zu 100 % beschleunigen! Das macht unser Speedchip! Kein Löten, Kinderleichter Einbau! Gratisinfo: M. Meyer G. Rohlfs-Str. 54c 282 Bremen 70 Tel.: 0421/6589278

BIETE SOFTWARE

Wegen Aufgabe: Megamax C 350,–, GfA-Basic V2.0 80,– DM, K-Seka Assembler 80,–; Tel.: 08024/2064

Neuw. Multi Mate Profi Textverarbeitungsprogramm von Ashton Tate Neuw. DM 2000,– VB DM 1200,– Unben. IBM BTX Adapter für Personal Computer Neuw. DM 1000,– VB DM 700,– Tel. 08821/71731

Haushaltsprogramm Monochrom
mit vielen Features DM 42,–
Info gegen Freiumschlag
A. Fisch, Büchner Str. 16,
2059 Bröthen

Riesiges PD-Softwareangebot, ab 3 DM pro Disk. Weit über 100 PD-Disketten, auch für den MacALADIN. Gratiskatalog bei WIRO-SOFT, B. Will & H. Rode Gbr, Langreder Str. 1, 3 Hannover 91, Mailbox 0511/735475 Menü .ATAS

DBMan GemVers 3 von M & T original versiegelt 300 DM; Tel.: 06204/2725

★ **ST-STATISTIK** unter **GEM** mit ★ Update-Service: deskrip. Daten para+nonparametr. Tests, Korr.-tabs, Varianz-, Faktor-, Cluster Analysen/Übernahme in Textprg. DM 249,– /DemoDisk DM 30,– Scheck/NN Dipl.Psych. M. Prall Isestr. 57 2 Hamburg 13

★ST-Software zu Superpreisen★
Flight II – 109 DM, SDI – 79 DM, Airball – 75 DM, Phantasie – 65 DM, Road-Runner – 69 DM, MusicStudio – 99 DM, V.D.Soft/V.Danner/Caspar-Baur-Str. 29/4230 Wesel
Kostenlose Info anfordern!

Kleine Anzeige – Kleine Preise
Sie werden **staunen!** Reichhaltiges **Soft- & Hardwareangebot** sowie über 300 PD-Disks! .Gratisinfo bei: **R. Markert, Balbachtalstr. 71, 6970 Lauda ☎ 09343/8269**

ST-MAIL Das Mailboxprogramm für Atari ST!! Info bei Software-Entwicklung und -Vertrieb - M. Nitschke, Bahnhofstr. 21, 6330 Wetzlar

---

★ ★ ★ ★ **Teacher's Friend** ★ ★ ★ ★
★ Praxisorientierte Noten- ★
★ verwaltung DM 69,– Gratisinfo! ★
U. Kunz, Junge Halden 3, 75 KA-1

Gratis Bei uns gibt es **Gratis**
Gratis schon PD-Software **Gratis**
Gratis gratis **Gratis**
Info bei: R. Markert, Balbachtalstr. 71, 6970 Lauda, ☎09343/8269

Public-Domain-Blitzversandt!
Riesenauswahl! Preise:
ab DM 4,– einseitige Disketten
ab DM 6,– doppelseitige
Inclusive Diskette !!!
Auch alle ST-Disks! Auf Doppeldisks beliebig kombinierbar!
Gratisliste anfordern bei:
A.Gauger Software,Buhlstraße 16a, 7507 Ettlingen, 07243/31828
(Bitte Computertyp angeben!)

■■ **Public-Domain DM 6,00** ■■
■ PD-Angebot dieser Ausgabe ■
■ **auf Mark.-Disk SS je DM 6,00** ■
■ Verp. DM 5,00, ab 20 St. frei! ■
■ V.-Scheck oder NN (+ DM 1,70) ■
■■ **Neu** ■■ **Neu** ■■ **Neu** ■■
■ **PD-Weihnachts-Pakete,** ■
■ interessant & günstig! ■
■ **Gratis-Liste anfordern!** ■
■ J. Rangnow/7519 Eppingen/Talstr. 8/07262/5131 ab 18 Uhr ■
■■■ **PD-Express-Versand** ■■■

SPITZENPROGRAMME FÜR ATARI ST! (freikopierbar) PAKET nur 45,– ● auch alle PD der ST-Computer bei Bestell. Katalog frei Haus durch ★ **hardsoft** ★ Inge Dander 089/3111782★ ★8 München 45
●●● Rockefellerstraße 66 ●●●

PD-Grafik-Pictures für ATARI-ST von Degas, Neo u.a. Katalog: Frey, Rheinstr.12A, 6538 Münster-Sarmsheim

---

★ Schweiz ★ Public-Domain-Software aus dieser Zeitschr. Info bei PD-Soft, Postfach 8, 8602 Wangen

---

FIBU, Buchführung für Freiberufler, Vereine, Privat, viele Funktionen (MWSt); schnelle Floppyversion, einf. Bedienung. Nur 59,– DM!!!
FAKTURA, Lager- u. Kundenverwaltung, Rechnung: 49,– DM.
HANDEL = FIBU+FAKTURA: 98,– DM.
MITGLIEDER-Verwaltung für Vereine u. Sonstige: Serienbrief, Beitrag, Mahnen, Abbuchen, Statistik etc. Einfache Bedienung. Nur 69,– DM!!! Info: Ottmar Kuschek, Mühlheimer Mühle, 5378 Blankenheim, 02449/1003.

LEERDISKETTEN? Tel: 0431/569216
Z. WOCHENENDE: Tel. 0481/72839

BS-Handel, Original Bavaria-SW, ROM-Modul etc., wegen Systemwechsel 390,– DM. 02449/1003.

PD-SOFTWARE, direkt vom Autor, die es **nur bei mir** gibt. **Riesen Angebot, alle Prg. einzeln** auswählbar. Der absolute Hit:
NUR 1,5 PF PRO KBYTE!! Mit super Tauschangebot für alle PD-Programmierer! 20-seitigen (!) **GRATIS**katalog bei Arne Zingel, Vermehrenring 11a, 24 Lübeck 1

ATARI ST-LOTTOSTATISTIK
39 DM, 20-seit. Dokumentation vorab gegen 2,40 in Briefmarken.
F.Klein, Dessauer Str. 5a, 89 Augsburg 21, Tel.: 0821/812157

■■ ★ ★ **GFA-Utilities** ★ ★ ■■
■ 1. Top-Quicksort: 489 Dat/sec. ■
■ 2. SUPER-Menu je Disk 23 DM ■
■ weitere kostenlose INFOS bei: ■
■ Tobias Winter / Hauptstraße 14 ■
■ 3112 Ebstorf ★ ★ ★ 05822/3941 ■

Achtung Musiker!!
Programmiert eure Drum-Maschine mit dem ST SQ-DRUM 99,–
Demo 10,– Info gratis:
Johannes Quack, Leineweberweg 3, 6200 Wiesbaden

★ ★ ★ **PD-Service** ★ ★ ★
Angebot wie PD dieser Ausgabe
Einzeldiskette SS DM 6.00
Doppeldiskette DS z.B. 1/2... 81/82 u.s.w. (aufeinanderf. u. ungerade beginnend DM 9.00
Pakete 1–10, 11–20, 21–30, 31–40 und so w. je DM 38.00 auf je 5 Disketten DS Preis inkl. Disk. zzgl. Port u. Verp. DM 5.00 (Ausl. DM 10.00) NN plus DM 1.70 besser V-Scheck – Lieferung erfolgt sofort, Neuheiten ab ca. 6. des Monats
I K S, Schönblickstr. 7
7516 Karlsbad – MU
ab 18 Uhr 07202/6793

FREDDIE!
Das Lernsystem für den Atari ST!
Voll unter GEM, extrem einfache Anwendung – Geballte Leistung!
Intern. Zeichen, viele Modi!
Info: M. Meyer
G. Rohlfs-Str. 54c – 282 Bremen 70

Verkaufe Omikron-Basic Modul!
Neu: 229,– DM! Bei mir:
≤ 150,– DM! Inclusive Handbuch, alles Original! Tel.: 0421/6589278

Suche Beschreibung für TDI-MODULA
☎ 04349/554 abends

Salix-Prolog DM 98,– – 030/6856960

Thermoschreibmaschine
Sharp-PA 1000, 8k, RS 232 – Interface, Anpassung 1st-Word, DM 300,–, Tel.: 08024/2064

EUSAX PD-Service 4018 Langenfeld
Haus Gravener Str. 85
★ RRG frei auswählen ab 4,50 DM ★
★ Kopie ab 2 DM nur Markendisk ★
ss/DS im Abo. Katalog gratis

BTX-MANAGER Atari ST – Btx mit GEM steuern, drucken. Daten in EDV auswerten
DREWS EDV+Btx *921733# ≪29>
Tel. 06221/163323

★ ★ Data Becker Programme ★ ★
Beckertext, Datamat*, Textomat* u. Textdesign* (*mit Sich.-Disk) VB: DM 450,– / Tel.: 06526/502

PD: EINZELKOPIEN und KOMPLETT-DISKS !! Liste gegen Rückporto:
D. Metz; Dyckburgstr. 18; 44 Münster

---

assoziative DATENBANK
THEMADAT schnelle assoziativ arbeitende **DATENBANK** 49,– DM, mit **TEXTMASKENGENERATOR** 74,– DM plus 5,– DM
Porto (V-Scheck) H.v.Tryller
3200 Hildesheim, Steinbergstr. 6

Stabwerk- u.v.a. Statik-Progr. sowie über 60 Mathem. Progr. Info (1 DM) bei: Jürgen Bullmann, Zur Hindenburgschleuse 3, 3000 Hannover 71

■ dBMAN dBMAN dBMAN ■
■Softw. + Bücher Info 17—21 Uhr
■SMC H. Keseling Fanny-Lewald-Ring 5, 2050 Hmb 80, 040/7351931

Wärmebedarfsberechng DIN4701 KZahlberechnung DIN4108 Wärmeschutznachweis WäSchV Dipl.Ing.V.Koch, Am Mehnacker 11 3563 Dautphetal 3, Tel. 06468/7652

ATARI-ST-Software in großer Auswahl ab Lager lieferbar. Kostenlose Preisliste anfordern. HEINICKE-ELECTRONIC. Kommenderiestr. 120, 4500 Osnabrück, Tel. 0541/82799. Tx 944 966

■ BS TIMEADRESS zu verkaufen originalverpackt, DM 190,— N.Fischer, 0041/41/458369 (CH)

PD-Software
★ Kopiergebühr: DM 1,—/Disk ★
★ Nur DM 0,50/Programm u.v.m. ★
★ Gratiskatalog anfordern bei ★
★ K. Galz, Sonnenbergstr. 55 ★
★ 78 Freiburg, Tel.: 0761/60367 ★

PD-Software
Großes Angebot an PD's u.v.a. ist vorhanden! Gratiskatalog bei L. Blenk, 2000 Hamburg 50 Ottenser Hauptstr. 65

PUBLIC DOMAIN
wie noch nie
Egal ob Spiele, Anwendungen, Sounds, Demos oder Dia-Show!
WIR MACHEN'S MÖGLICH

Hans Frey
Don-Carlos-Str. 33 B
7 Stuttgart 80

SUCHE SOFTWARE

Suche frivole Software
Tel.: 06109/62186

COBOL-COMPILER gibt es sowas für den ST? Bitte Angebot an Dr. G. Hlawatsch, c/o PGT-Europa GmbH, Mainzer Str. 103, 6200 Wiesbaden, Tel.: 06121/719052-55

SUCHE HARDWARE

LIGHT PEN für ST? Wer kann uns helfen 030/3003478 oder 3003487

Suche SM 124, Preis ca. 180,— DM K. Hiller, In der Heufahrt 1, 6233 Kelkheim, 06174/62890

ST 1040 ges. 08226/599 (abends)

Student sucht defekte HARDDISK für ATARI zu günstigem Preis Tel.: 06126/2705

TAUSCH

Von Lehrer für Lehrer! Notenverw.prg. Markbook plus in GEM. Über 20 versch. Funkt. f. alle gäng. Notensyst. Spez. vers. auf Wunsch; Info gegen Freiumschlag. H.J.Merkel/Nahestr. 28/6600 SBR

★ ★ ★ Schule und Lernen ★ ★ ★
Lehrer sucht Tauschpartner mögl. Raum Stuttgart. Atari ST und C-64 Werner Knobloch, Auf der Stelle 27, Tel.: 07031/804045 in 7032 Sindelfingen

KONTAKTE

Suche ST-Kontakte und Tauschpartner in Delmenhorst/Bremen 0421/50694 nach 18 Uhr — Gerd

★ ST-SOFTWAREENTWICKLUNG ★
Schreibe Software, die speziell auf Ihre Bedürfnisse zugeschnitten ist. Kostenl. Info anford. Thomas Levfkes, Knappenstr. 10, 4353 Oer.-Erkenschwick Tel.: 02368/55020

'C'-Porgrammierung!!! Suche Leute zum gemeinsamen profess. Einstieg in C u Ass W. Lampert Hoistener Str. 35, 404 Neuss 476130

VERSCHIEDENES

★ 10 Neue Fonts für SIGNUM ★
★ Info und Probedrucker von: ★
★ H. Manschus, Tel.: 0651/76814 ★
★ Karl-Marx-Str.-70b, 55 Trier ★

Prof. Soundunterstützung durch digitalisieren von Sounds, Effekten, Sprache !! Tel.: 02529/1496 !!

Suche „ST-Computer"-Sammlung, 1/86—6/87 (kompl.) 0241/402617

Bevor Sie in DTP investieren, sollten Sie unser Angebot kennen! Wir lesen Ihre Disk, erst. Druckvorlagen, drucken m. Laser u. Offset. Tel.: 06021/75791

Anzeigenschluß für ST-Computer Heft 1/88 ist der 13. 11. 1987



Der exclusive Service der Redaktion

Public Domain Disk Nr. 100

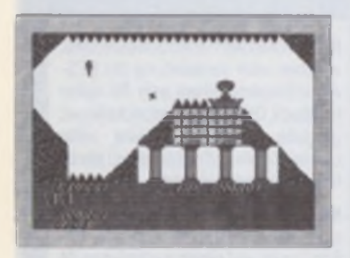
Es freut uns, daß wir Ihnen diesen Monat unter anderem die hundertste PD-Diskette präsentieren können. Wir hätten nicht gedacht, daß dies schon dieses Jahr geschehen würde.
Aus diesem Grund möchten wir uns bei allen Autoren bedanken, die diese Sammlung mit unterstützten. Wir wissen um die Arbeit, die in diesen programmen steckt, wissen aber auch, daß der Hintergrund einfach die Freude am Programmieren ist. Doch auch den Benutzern dieser Programme sei gedankt, denn auch Sie haben mit Fragen und Anregungen dazu beigetragen, daß viele Autoren ihre Programme noch wesentlich verbesserten und uns wiederum Updates schickten, über die wir Sie informieren. Auch unsererseits ist dieser Service eine aufwendige Sache. Das Zusammenstellen der Disketten nach bestimmten Kriterien beispielsweise (Utilities, Anwender, Spiele, ...), die Überprüfung der Programme auf Fehler und vor allem auf Qualität, denn nur gute Programme werden aufgenommen, sonst gäbe es schon 200 Disketten. Trotz dessen, oder gerade deswegen, sind auch wir ein wenig stolz auf diese Sammlung, denn sie ist sicherlich die einzige sortierte dieses Umfangs. Wir möchten daher alle Programmierer auffordern sich mit neuen Programmen und neuen Ideen an der ST-Computer Public-Domain Sammlung zu beteiligen um die Sammlung noch reichhaltiger zu gestalten.

Ihre ST-Computer Redaktion

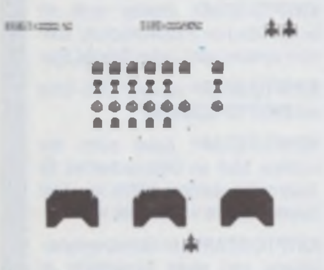


Spielesammlung

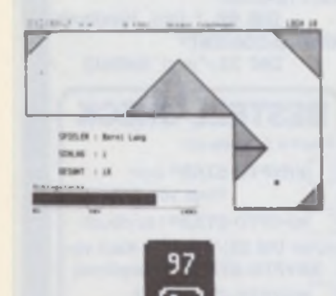
- ***Andurin:*** Ein Geschicklichkeitsspiel mit 75 Bildern. Die Aufgabe ist es verschiedene Räume zu durchqueren, Gegenstände einzusammeln und Hindernisse auszuweichen. (s/w)



- ***Gilgalad:*** Arcade Adventure mit 224 (in Worten zweihundertvierundzwanzig) Räumen. (s/w)
- ***Invadors:*** Der Spieleklassiker. Außerirdische Raumschiffe greifen die Erde an. Ausdauer und Timing ist gefragt. (s/w)



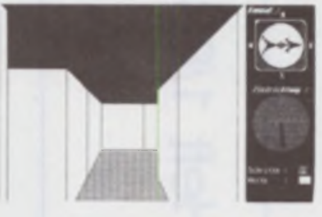
- ***Minigolf:*** Wem das Spiel in natura zu anstrengend ist, der kann es auch auf dem Rechner spielen. 18 Bahnen, Steuerung (Richtung, Anschlagstärke) per Maus (s/w)



97

Diskette 97

- ***Trash:*** The First Trash-Groove Adventure. Ein Textadventure mit **deutschen** Befehlen. Der Inhalt ist eine Parodie auf die Musik- und Punkszene. Sie müssen in einer verwahrlosten Stadt ihren entführten Hund wiederfinden und erleben so allerlei verrückte Dinge. (incl. Source in Omikron.-Basic)
- ***Labyrinth:*** In einem variable großen Labyrinth auf der Suche nach dem Ausgang. Sehr schöne 3D-Darstellung. 2D-Darstellung als Hilfe. Schnell und druckfrei. (s/w)





98

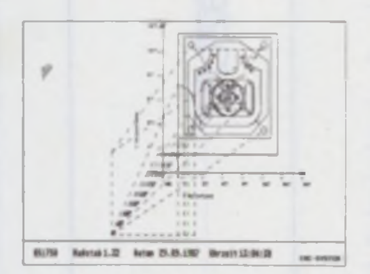
VAX GAMES

- ***SNAKE:*** Eine Schlange auf Nahrungssuche. Vorsicht vor dem eigenen Schwanz
- ***QIX:*** Sie müssen 75 % des Bildschirms einzäunen ohne gebissen zu werden (ähnlich STIX)
- ***Nibbler:*** Eine Schlange schlängelt sich durch ein Labyrinth
- ***Pacman:*** Kommentar überflüssig
- ***Doors:*** Pacman Variante mit Hindernissen
- ***Robot:*** Das Spielfeld muß leergeräumt werden ohne die ziemlich aufdringlichen Roboter zu berühren.

99

Diskette 99

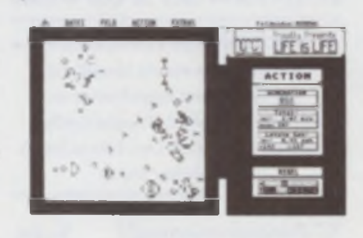
- ***CNC Simulation:*** Dieses Programm simuliert eine CNC gesteuerte Fräsmaschine. Programmieren Sie ihr eigenes Fräsprogramm; und lassen Sie es auf dem Bildschirm simulieren. Vielseitige Befehle, eben wie eine CNC-Maschine. (s/w)





100

- ***ST-Speech:*** Das gab's bisher noch nicht! PD-Sprachausgabe auf Softwarebasis. Das Programm spricht jeden eingetippten Satz. Wandelt englischsprachige Texte automatisch in verständliche Lautschrift. (s/w)
- ***Fortran Shell:*** Eine Shell für Fortran 77. Die Tipperei hat ein Ende. Bequemes Aufrufen von Compiler, Linker, und anderen Programmen aus einem Menü heraus
- ***Life is Life:*** Dieses Simulationsprogramm ist an Komfort wohl kaum noch zu überbieten. Sechs verschiedene Feldgrößen, Drehen, Verschieben, Kopieren von Blöcken wie im Malprogramm, Festlegen der Fortpflanzungsregeln, hohe Geschwindigkeit. (s/w)

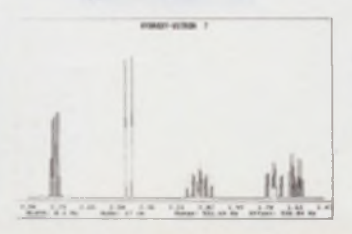


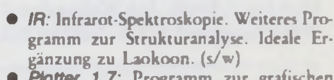
- ***Fastlife:*** Life-Simulation in Volksforth. Auch wenn nicht so komfortabel zu bedienen, so ist dieses Programm das schnellste was je da war. Zig Generationen werden in einer Sekunde berechnet und angezeigt. (s/w)
- ***GEMFRAC:*** Berechnung fraktaler Landschaften in Farbe und s/w. Viele Möglichkeiten im GEM-Gewand
- ***Sharp:*** Sammlung vieler Basic Programme für Sharp Basicrechner. Geeignet für Direktüberspielung (mit Interface) oder einfach zum Abschauen.



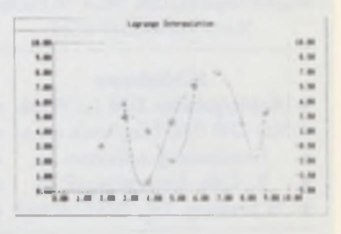
Wissenschaftliche Anwendungen (Schwerpunkt Chemie)

- ***Laokoon:*** Simulation von NMR (Magnetische-Kern-Resonanz-Spektoskopie). NMR ist in der Chemie eine weit verbreitete Methode zur Strukturaufklärung. Bestimmung von chemischen Verbindungen und Kopplungsarten. (s/w)





- ***IR:*** Infrarot-Spektroskopie. Weiteres Programm zur Strukturanalyse. Ideale Ergänzung zu Laokoon. (s/w)
- ***Plotter 1.7:*** Programm zur grafischen Darstellung von Werten, die z. B. aus Messungen stammen, als Kurvenzüge. Dabei können sie entweder als normale- oder Lagrange Interpolation, Spline oder Regressionsgrade ausgegeben werden. Berechnung des Korelationskoeffizientes. Automatische Achseneinteilung, Angabe des Zahlenformats (Vor- und Nachkommastellen), Beschriftung. (s/w)

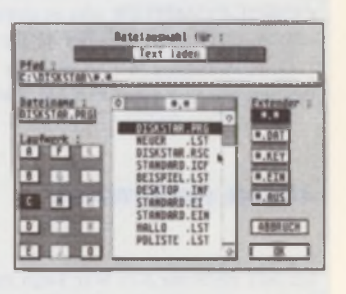


- ***Plottit:*** Kurvenplotter von mathematischen Funktionen. Direkte Funktionseingabe mit sofortiger Syntaxprüfung und Anzeigen einer eventuell fehlerhaften Stelle. Nullstellenberechnung. (s/w)
- ***Ganglion:*** Programm zur Simulation Neuraler Netze. Sieh Artikel 'Künstliche Intelligenz, Teil 7'. (s/w)



Programmierutilities

- ***Fileselect:*** Eigene Fileselectroutine. Wesentlich flexibler als die original ATARI Fileselectorbox. Zum Einbinden in eigene Programme



- ***Printf:*** Verbesserte PRINTF-Routine für Megamax C
- ***RCS_ICN:*** Programm zum Ausschneiden beliebig großer Bildschirmbereiche und Umwandlung derselben in ICONS. Es lassen sich damit sehr leicht neue ICONS erstellen.



Public-Domain Service

Der Service für unsere Leser

Die Disketten 1...38 sind auch weiterhin lieferbar. Schauen Sie dazu in eine der älteren Ausgaben der Zeitschrift, oder fordern Sie die Liste an (frank. Rückumschlag)

40. Diskette
● *Deluxe Fontmaster ST*: Druckprogramm zum Erstellen von Karten und Tabellen, mit vielen Zeichensätzen, Proportionalschrift und Blocksatz, eigener Text- und Zeichensatzeditor. Bis zu 10 Schriftarten in einem Text mischen und vieles mehr [s/w]

41. Diskette
● *Painter*: Zeichenprogramm unter GEM. Verfügt über einige Besonderheiten. Z. B. Zoom, Linien und Füllmustereditor, Blockoperationen (Verschieben, Drehen,...) [s/w] ● *VIKA*: Bedrucken von Kassetten und Videohüllen [s/w] ● *Kalender*: Diverse Kalenderberechnungen. U. A. Ausgabe eines Jahreskalenders auf Drucker oder Bildschirm, mit Anzeige aller Feiertage (sehr wichtig)

42. Diskette
● *FRAM 77*: Statische Berechnungen für Konstrukteure. Sehr umfangreich ● *Meßwert*: Grafische Darstellung von Meßwerten [s/w], incl. Sourcecode in Pascal

43. Diskette
● *System*: Accessoir ähnlich dem Kontrollfeld. Besonderheiten: Uhr und 4 Weckzeiten, Druckereinstellung (Rand, Schriftart,...) [s/w], incl. Sourcecode in Pascal ● *Snaphelp*: Hiermit kann SNAPSHOT (1st Word Utility) auch auf DEGAS angewendet werden, incl. Sourcecode in Pascal

44. Diskette
Spieledisk
● *TTT*: Dreidimensionales Tic Tac Toe (4 * 4 * 4) [s/w] ● *El Bozo*: El Bozos Window Puzzle ● *Senso*: Fordern Sie Ihr Gedächtnis [s/w] ● *Kalah* ● *Tri Hex*: Räumliches Tic Tac Toe (3 * 3 * 3) [s/w] ● *Börse*: Spekulieren am Börsenmarkt [s/w] ● *Solitär* [s/w]

24h-Service
(Bearbeitungszeit)
Anruf genügt

45. Diskette
● *Complex*: Bibliothek f. komplexe Berechnung ● *Master*: Generationsberechnung mit frei definierbaren Regeln [s/w] ● *Pegasus*: Malprogramm [s/w] ● *Con_ST*: Terminalprogramm u. a.

46. Diskette
● *Termin*: Terminplanung als Accessoir [s/w] ● *KFZ-Stat*: Statistikprogramm für's Auto [s/w] ● *Lager*: Lagerverwaltung [s/w] ● *ST_Phone*: Telefonkostenberechnung [s/w]

47. Diskette
Utilitydisk
● *M_Copy_87*: Backupprogramm mit vielen Optionen und Musik ● *Copy_FMT*: Backupprogramm als Accessoir ● *Uhrzeit*: diverse Utilities zur eingebauten Uhr ● *1st_step*: Kalender und Biorhythmus [s/w] ● *Lotto* [s/w] ● *P2_init* u.a.

48. Diskette
● *Word 1.4*: Programm zum Erstellen eines luxuriösen Lexikons für 1st Wordplus

49. 50. 51. Diskette
● *Volksforth 83*: Die neueste Version des Forthinterpreters. Die Software zum Forthkurs. Alle drei Disketten gehören zum Forth-System.

52. Diskette
● *Molekül*: Darstellung chemischer Moleküle. In s/w und Farbe, wobei in Farbe mit 3D-Effekt. Atom, Ionen oder Van der Waalsradius. Ein Molekül kann gedreht und sogar animiert werden.

53. Diskette
● *Matrix*: Komfortables Matritzenberechnungsprogramm mit voller GEM-Steuerung (Icons, Windows, Menüzeile). Maximal 16 * 16 Matritzen, doppelte Genauigkeit (64 Bit). ● *LIB*: Sammlung von ST-Pascal-Bibliotheken. Incl. Source und Anleitung. Stringumwandlung (str <-> int, long_int, real) Datums + Uhrzeit-Rechnungen Textformatierungen (links, rechtsbündig, zentriert, u.v.m.)

54. Diskette
Spielesammlung
● *Bounce*: Bouncing Bubbles, Schnelles Actionspiel für Joystick und Tastatur (s/w) ● *Domino*: Gefräßige Würmer auf Nahrungssuche (s/w) ● *Trio*: Verbessertes Tic-Tac-Toe mit Verschieben (s/w) ● *Disc*: Die Serie der Spielautomaten reißt nicht ab (s/w)

55. Diskette
Editorendisk
● *TOSEDIT*: Editor unter TOS. Gleichzeitiges Bearbeiten von zwei unabhängigen Texten. Reichhaltige Befehle. ● *EDIMAX*: Editor unter GEM. Mit Funktionstastenbelegung. Weiterhin zeichnen sich EDIMAX durch sehr schnelles Scrolling aus.

56. Diskette
● *Ausland*: Fremdsprechentrainer Englisch mit Sprachschatz für Fortgeschrittene. ● *Fugger*: Verwalten Sie ein Fürstentum und versorgen Sie ihr Volk. (s/w) ● *Yahtzee*: Altbekanntes Zockerspiel auf den ST umgesetzt.

57. Diskette
Wissenschaftliche Anwendungen
● **Kepler**: Grafische Darstellung von Himmelskörper-Bewegungen nach den Keplerschen Gesetzen. Variation von Masse, Radius, Zeit, potentieller und kinetischer Energie. ● *Kerne*: Der Krieg der Kerne, eine futuristische Schlacht; im Innern eines Computers suchen Rechenprogramme sich gegenseitig zu zerstören. ● *Meßwert*: Verarbeitung von Meßergebnissen und grafische Darstellung (s/w).

58. Diskette
● *SM-Konverter*: Konvertiert SM-Manager-Texte in 1st Word (Wordplus)-Texte und natürlich auch zurück. ● *Compress*: Kompressiert beliebige Files. ● *Label*: Ändern des Diskettennamens. ● *Eti-Master*: Etikettendruck zu DB-Master ● *Urlaub*: Hilft den Urlaub zeitoptimiert zu berechnen.

59. Diskette
● *Themadat*: Assoziative Datenbank mit bis zu 200 Datensätzen.

60. Diskette
● *Synthi*: Erzeugt div. Sounds, die abgespeichert und über 'DOSOUND' in eigene Programme übernommen werden können (incl. der Source-Code) (s/w) ● *Waschi*: Waschhilfsprogramm für Junggesellen (s/w) ● *Expert*: Ein einfaches, lernbegieriges Expertensystem ● *Adressen*: Adressverwaltungsprogramm mit Etikettendruck und einfacher Textverarbeitung

61. Diskette
● *Hacomini*: Erzeugt Hardcopies in Miniaturformat (48 mm * 68 mm) (s/w) ● *Helper*: Hilfsprogramm bestehend aus Diskmonitor und Druckerutility ● *Bidipo*: Scrollt eine Bilderkette über den Schirm. Ideal für eigene Diashows oder Werbegrafiken ● *Datobert*: Grafische Darstellung. Linien- und Kuchengrafik, sowie in dreidimensionaler Balken- und Flächengrafik. Überlagerung einzelner Datenserien (s/w) ● *Direct all*: Directorydruck ● *Überweisung*: Überweisungsdruckprogramm aus ST 6/87.

62. Diskette
Midi-Diskette
● *MIDIDRUM*: Rhythmusmaschine mit leicht und vielseitig editierbaren Sounds ● *MIDIPLAY*: Erlaubt das Einspielen, Speichern und Ausgeben von Midi-Sequenzen, incl. Source-Code in GFA-Basic.

63. Diskette
Spielesammlung
● *Fuffy*: Sehr schön gestaltetes Kniffel-Spiel mit zusätzlicher Auswertegrafik (s/w) ● *Snake*: Eine Schlange schlängelt sich durchs Leben, auf der Suche nach Fressen. Man achte auf Hindernisse (s/w) ● *Bomber*: Scramble-Spiel (f) ● *Reversi*: Der Name sagt alles (f).

64. Diskette
● *E-PLAN*: Konstruktionsprogramm für elektronische Schaltungen. Alle elektronischen Bauelemente sind vorhanden. Schaltung können auch beschriftet und die Bauteile in die Stückliste übernommen werden.

65. Diskette
Mal- und Zeichenprogramme
● *Columbia*: Malprogramm mit vielen Extras. Mehrere Bildschirme, Blockoperationen, Spiegeln, Verzerren, Drehen, Lupe (s/w) ● *Paintlux*: Malprogramm mit zusätzlicher Bibliotheksverwaltung. Dadurch können z. B. elektronische Bauteile aus der Bibliothek in das Bild eingefügt werden (s/w).

66. Diskette
Spieledisk
incl. Source-Code in OMIKRON Basic
● *MAZIACS*: Abenteuerspiel. Unser goldsuchender Held im Labyrinth, im Kampf mit bösartigen Spinnen und anderen Hindernissen (s/w) ● *Quadrat*: Bilderpuzzle. Ordnung im Chaos (s/w).

Diskette 67
Utilities
● *Diskchecker*: Prüft eine Diskette auf Fehler. Formatier- sowie Schreib- und Leseprüfung (s/w). ● *DEGAS ELITE Fontmaker*: Erzeugt das DEGAS ELITE-File Assign.SYS. Damit wird die Einbindung neuer Fonts stark vereinfacht. ● *PRINT-SET*: Druckeranpassungshilfe. Sonderzeichenausgabe. ● *TESTBILD*: Das Farbmonitor-Testbild. ● *ICONEDIT*: Sprite-, Maus- und Füllmuster. Übernahme in eigene Programme.

Diskette 68
Mal- bzw. Zeichenprogramme
● *Studio*: Zeichnen in DIN A4-Format. Blockverwaltung. Automatische Bemaßung. Ausschnitts-Hardcopy. ● *Van Gogh*: Mal- und Animationsprogramm. 64 bewegte Blökke. Incl. Blockeditor (f).

Diskette 69
● *Astrolabium*: Zeichnen von Sternkarten- oder Himmelsausschnitten, zu beliebiger Zeit, an beliebigem Ort. Einstellbare Blickrichtung, -höhe und -winkel. Incl. Daten von ca. 9000 Sternen (s/w).

Diskette 70
● *Orbit*: Zeigt die Laufbahn von Satelliten als bewegte Grafik über der Weltkarte. Z. B. für Funkamateure (s/w). ● *High End*: Programm zum Berechnen von Lautsprecherboxen aller Art. Lautsprecher und Boxenmaße. Datei für Speaker enthalten (s/w).

Diskette 71
Spieledisk
● *Ballerburg*: Zwei verfeindete Burgherren zerstören sich gegenseitig ihre Burgen. Steuerung der Kanonenwinkel und der Pulvermenge (s/w). ● *Stagger*: Reversiumsetzung (s/w). ● *Hotelier*: Hotelverwaltungsspiel. Leiten Sie Ihr Hotel und kämpfen Sie gegen die Konkurrenz (s/w). ● *Roulette*: Was mag dies wohl sein? (s/w)

Diskette 72
● *ZREF*: Crossreferenz für C-Programme. Erzeugt Variablenliste mit Funktionsaufrufen, Typen, defines, structs, Zeilennummern und vielen anderen Optionen. ● *Plotter*: Werteplotter für wissenschaftliche Anwendungen. Lineare-, Lagrange-, 3-Punkte-Interpolation sowie Regressionsanalyse. Nachträgliches Editieren (s/w). ● *Nicelist*: Ausdruck von Source-Codes mit Seitennummern, Dateiname und Datum. Sonderzeichenanpassung.

Diskette 73
● *Videoarchiv*: Videokassettenverwaltung. Leicht bedienbar. Mit Etikettendruck (s/w). ● *Lohnsteuer*: Lohnsteuerberechnungsprogramm. ● *Hausmanager*: Kontrolle über die private Haushaltsabrechnung. Jahresübersicht (s/w).

Diskette 74
Ramdiskpaket u.a.
● *MAXIDISK*: DIE Ramdisk. Resetfest (Inhalt nach Reset noch vorhanden). Komprimiert ihren Inhalt (eine 500K-Ramdisk kann wesentlich mehr aufnehmen als 500K). Frei wählbare Größe, frei wählbare Laufwerkskennung ● *COPY*: MAXIDISK-Utility. Kopiert Inhalt eines Ordners beim Systemstart in die Ramdisk. ● *TIME*: MAXIDISK Utility zum Stellen des Datums und Uhrzeit. ● *Mono Neo*: Speichert bei Alternate + Help den Bildschirm auf Diskette. ● *Autocopy*: Ähnlich dem Programm COPY. Kopiert bestimmte Dateien in beliebige Ordner. ● *Datebook*: Erinnert beim Systemstart an wichtige Termine oder Geburtstage. u.a.

Diskette 75
Fontdiskette zu Fontmaster (PD40): Neue bzw. erweiterte Fonts und Rahmen. Ergänzung zu Fontmaster (PD40).

Diskette 76
Schulungssoftware
● *Periode*: Das Periodensystem der Elemente. Chemielernprogramm mit vielen Optionen. Enthält die Daten jedes chemischen Elements (s/w). ● *Statistik*: Grafische Darstellung und Datenauswertung (s/w).

Diskette 77
Utilitydisk
● *FCOPY 2.0*: Schnelles Kopierprogramm. Formatieren, Verify, Multiple Copy, Scan und Teilkopie. ● *Speeder*: Verdoppelt Schreib- und Lesegeschwindigkeit. ● *File-copy*: Kopieren von Einzeldateien. Auch mit einer Floppy. ● *Formater*: Erhöht die Diskkapazität. ● *Fastformat*: Formatiert Schnelladedisks. Durch besondere Anordnung der Sektoren wird immer schneller geladen. ● *Taskcopy*: *Diskbackup im Multitasking* ● *Profiramdisk*: *Resetfeste Ramdisk mit Ausführungsprofil*

Diskette 78
Utilitydisk
● *AESLIB*: Bibliothek für GFA-Basic. Enthält komplette AES-Bibliothek. Mit Linker, der jede benötigte Prozedure automatisch zum Programm hinzulädt. ● *LINE_A*: Line_A Binding für C-Programme. ● *Conv.WP*: Konvertiert ASCII-Dateien in den WP-Code von 1st Word. Dadurch Umbruch, Formatieren etc. möglich. ● *Player*: Programm zum Abspielen von Interruptmusik incl. Source in C. ● *Musik*: Ähnliches Programm aber als Accessoir. ● *Setz_Uhr*: Programm zum Stellen der Zeit + Datum incl. Source in Assembler.

Diskette 79
Etikettendruck
● *Label Expert*: Druckt alle Art von Etiketten. 3,5 Zoll- und 5,25 Zoll-Disketten, Audio-Kassetten, Video-Kassetten, Ordner u.a. (s/w). ● *Grafikett*: Bedruckt Disklabels auch mit Grafik, die aus beliebigem Grafikbild ausgeschnitten wird (s/w). ● *Etikettdat*: Druckt Adressenaufkleber (s/w). ● *Etikette GFA*: Luxuriöses Disklabeldrucker. Mit vordefinierter, artbezogener Grafikeinbindung (s/w).

Diskette 80
Spieledisk
● *U_Boot*: Ziel dieses Spiels ist es, U-Boote im Meer mit eigenen Sonden zu orten und ihnen auszuweichen (s/w). ● *Ärger*: Eine Umsetzung des wohl bekanntesten Brettspiels 'Mensch ärger dich nicht' auf den ST (s/w).

Diskette 81
Malprogramm
● *Public Painter*: Malprogramm mit interessanten Fähigkeiten. Verarbeitet z. B. gepacktes Format oder Bilder von SNAPSHOT. Eingebaute Lassofunktion zum Ausschneiden beliebiger Teilbereiche. Blockmanipulation ähnlich EASYDRAW. Vergrößern, Verkleinern, Drehen, Biegen, Verzerren, Überschneiden. Drucken von Bildausschnitten (s/w).

Diskette 82
Malprogramm
● *Masterpainter*: Läuft in allen drei Auflösungen. Konvertiert jede Auflösung in beliebige andere. Blockbearbeitung. Vergrößern, Verkleinern, Rotieren, etc., bis zu 24 Screens. Ausschnittdruck ● *Little Painter*: Malprogramm. Rasteroperation, Musterspray, Verwischen, Drehen, Biegen, Blockladen und -speichern, Fontlader.(s/w)

Diskette 83
● *Starofix*: Zeichsatzeditor für Down-Load Zeichen. Star NL10 und kompatible. Zeichensatz per Autostartprogramm in Drukker laden. Viele Fonts ● *Starlist*: Ausdruck von ASCII-Dateien mit Sonderzeichenwandlung.

Diskette 84
Druckedit: Zeicheneditoren für 1st Word Druckeranpassung ● *Draft Pin*: Zeichen werden editiert und automatisch in beliebiges .HEX-File eingebaut. (s/w)

Diskette 85
Spieledisk
● *Skat*: Eine Runde SKAT gegen zwei von ST gesteuerten Zocker. (s/w) ● *Delta*: Gehirnmuskeltraining. Ähnlich dem Zauberwürfel. (f) ● *Gruselt*: Grafik- und Sounddemo. Geisterstunde auf dem Friedhof. Skelette, Abte und Gespenster treiben ihr Unwesen. (f)

Diskette 86
● *Carpet*: Berechnungsprogramm von 3D-Funktionen. Mit Hidden-Line Algorithmus und mit Beleuchtungseffekten. Funktionseingabe und frei einstellbare Parameter. (s/w) ● *3D-Plot*: Ähnliches Programm in compiliertem GFA-Basic. Obwohl die Programme ähnlich sind, so bietet doch jedes seine Spezialfunktionen. (s/w) ● *Funktionsplot*: Funktionseingabe im Programm, verschiebbare Achsen, mehrere Darstellungsarten (s/w)

Diskette 87
● *Sternbild*: Zeigt den Sternenhimmel, oder bestimmte Ausschnitte zu verschiedenen Zeiten. Planetennamen auf Anklicken. S/W und Farbe). Im Farbbetrieb erzeugt es auch ein räumliches Rot-Grünbild

Diskette 88
Druckertreiber
● *NEC-EMU*: Der bereits in den NEWS angekündigte Hardcopytreiber für die NEC Px-Reihe. Erzeugt Hardcopies in 12 verschiedenen Größen und Kontrasten. Enthält Druckerspooler ● *LQ 800*: Ein Hardcopytreiber für den EPSON 24 Nadeldrucker in drei verschiedene Größen ● *NEC_Code*: Ein Programm zur Einstellung des NEC P6 /P7 u. ä. ● *SIMPLE*: Komfortables Hardcopy-Programm mit vielen Extras ● *STX-80*: Hardcopytreiber für diesen preiswerten Thermodrucker ● *WORDPLUS*: Druckertreibersammlung für folgende Drucker: NL10, FX1000, SP1000I ● *1st prop*: 1st Proportional Treiber für folgende Drucker: HR15, FX85.

Diskette 89
● *Führerschein*: Haben Sie schon den Führerschein? Dann haben Sie Glück. Falls nicht hilft Ihnen dieses Programm zumindest bei der Theorie. Ca. 500 Fragen sind enthalten ● *Bundesliga*: Zwar hat die Saison schon begonnen, aber entschieden ist ja wohl noch nichts. Diese Programme dienen zum Anlegen einer Tabelle, editieren aller Spiele von der 1., 2. Bundesliga, bis hin zur untersten Bezirksliga.

Diskette 90
Utilities
● *TOPS*: The Other Pascal Shell Erlaubt die Einbindung von Tempus, erzeugt Crossreferenz und enthält Befehlezum Kopieren, Löschen und Umbenennen von Files ● *SIGNUMShell*: Eine Shell zu Signum. Man kann nun per Menü zwischen den Signum Editor, den verschiedenen Druckprogrammen, den Zeicheneditoren hin und herschalten ● *Tempelmon 1.8*: Neue Version des weitverbreiteten Monitors. Läuft nun auch in Farbe ● *Maxidisk 4Mb*: Die komprimierende Maxidisk mit max. 4MByte Kapazität. Läuft auch auf Blittertos. Komprimiert automatisch ihren Inhalt. Resetfest ● *RAM-DISK*: Gleiches Programm wie MAXIDISK, nur ohne Komprimieralgorithmus. Beide Ramdisk enthalten ein Autocopyprogramm zum kopieren bestimmter Files in Ramdisk ● *RCS Wandel*: Wandelt *.h Files nach Basic und Assembler.

Sämtliche Disketten sind in dieser aktuellen Form nur direkt bei der Redaktion erhältlich. Für anderweitig bezogene Kopien können wir keinen Service leisten.

Diskette 91
Lernprogramme
● *Pauk*: Lerndatei mit komfortabler Steuerung, eigener Dateneingabe, Wiederholung falscher Fragen nach Lernkonzept. (s/w) ● *Hypervoc*: Sehr schön gestaltetes Vokabellernprogramm. (s/w) ● *ST-Dic*: Fremdsprachenlernprogramm mit schon vorhandener großer Lateinbibliothek ● *QUIZ*: Eine Art Trivial Persuite. Enthält Fragen mit drei möglichen aber nur einer richtigen Antwort. Eigene Fragen einbinden möglich. Enthält (ATARI ST-Einsteigerkurs).

Diskette 92
Utilitydisk
● *Superbatch*: Batch-Prozessor: bedingte Batchbearbeitung (if) / Batch Protokoll auf Bildschirm, Drucker, Datei / Verzweigung. Fehlerbehandlung ● *ST-Klick*: Multifunktionsaccessoir: Notizen, Wecker, Drucker Init, Schreibmaschine, Kalender, Taschenrechner (s/w) ● *XREF*: Crossreferenz zu C ● *Term*: Terminalprogramm mit Xmodem ● *Allaformat*: flexibles Formatierprogramm (9–11 Sektoren) (bis 86 Tracks).

Diskette 93
● *ADR_2* ● *Dat_Text* ● *R_Adress*

Diskette 94
Spieledisk
● *MacPan*: Eine schnelle Pacman-Version für den ST. Per Joystick ins Labyrinth auf der Suche nach Powerpills ● *Solitär*: Dreidimensionales Solitär mit sehr ansprechender Grafik ● *Galaxy Sprengmeister*: Brettspiele zur Förderung der Gehirnaktivität (s/w)

Diskette 95
Spieledisk
● *Diamond Miner*: Ein Spiel ähnlich Boulder Dash. Sehr schnelle Grafik und 30 Level. Eingebauter Feldeditor, bei dem nach Belieben eigene Spielfelder generiert werden können. (s/w) ● *Snake*: Zwei Schlangen versuchen sich den Weg abzuschneiden. (s/w) ● *Minenfeld*: Ein wagemutiger Schatzsuchen begibt sich durch verschiedene Minenfelder und muß versuchen die Minen zu umgehen. (s/w) ● *Missile*: Missileattack (s/w)

Sonderdisks

Die folgenden Programme sind nicht Public-Domain.
Sie können aber bei uns bezogen werden.

A.) – *TOS*: Die letzte Disketten-Version des TOS vom 6.2.1986. Zum fehlerfreien Betrieb des Rechners benötigen Sie diese Version, da z. B. mit älteren Versionen manche Ordner nicht gelesen werden (0 Bytes free). Unkostenbeitrag 15,– DM incl. Diskette.

B.) – *RCS*: Das Resource Construction Set aus dem ATARI-Entwicklungspaket. Unverzichtbar bei der Erzeugung von Objektbäumen, Dialogboxen, Drop-Down Menüs. Kurzum: Ein Muß für GEM-Programmierung. Unkostenbeitrag 15,– DM incl. Diskette.

Bis auf die Höhe des Unkostenbeitrags gelten die gleichen Versandbedingungen wie bei der Public-Domain-Software.

Versandbedingungen:

Sämtliche Disketten können Sie ab dem Erstverkaufstag direkt von der Redaktion erhalten. Wir haben für Sie den schnellstmöglichen Versandservice eingerichtet. 24h-Service – Bearbeitung innerhalb eines Werktages

1. Anruf genügt:

ST-Computer Redaktion
Tel.: 0 61 96 / 48 18 11
BRD / 61 96 / 48 18 11

2. Schriftliche Bestellung:

– Der Unkostenbeitrag für eine Diskette beträgt DM 10,–
Bezahlung per Scheck oder Nachnahme (Nachnahme nur im Inland).
– bei Nachnahme zuzüglich DM 3,50 Nachnahmegebühr
– Fügen Sie dem Betrag folgende Versandkosten (Porto + Verpackung) bei: Inland DM 5,– Ausland DM 10,–
– Legen Sie bitte, falls zur Hand, einen Aufkleber mit Ihrer Adresse bei.

Die Bezugsadresse lautet:

ST-Computer-Redaktion
MERLIN-Computer GmbH
'PD-Service'
Postfach 59 69
D-6236 Eschborn

Bei den Disketten handelt es sich um FUJI MF1DD, die wir uneingeschränkt für ein- und zweiseitige Laufwerke empfehlen.

Bei Fragen bezüglich der Disketten wenden Sie sich bitte an die Redaktion.

Clubs AUF EINEN BLICK

An alle Userclubfreunde

Häufig werden wir nach Clubadresen gefragt. Nicht immer können wir helfen, da es sogar in manchen Großstädten keine Clubs gibt, oder diese uns nicht benachrichtigen. Deshalb noch einmal der Aufruf an alle, hier nicht aufgeführten ATARI ST Clubs, uns ihre Adresse bekannt zu geben. Im Folgenden die bekannten Clubadressen:

D-2000 Hamburg
DELMONICO
Niklas Nebel
Schloßstr. 100
D-2000 Hamburg 70

D-2724 Sottrum
ACT
Atari Computer Team
Postfach 11 27
D-2724 Sottrum

D-2951 Nortmoor
STUCO
ST User Club Ostfriesland
Detlef Koppelkamm
Duesterweg 30
D-2951 Nortmoor

D-2960 Aurich
ST Club Aurich
Erik Dreesen
Hafenstr. 6
D-2960 Aurich

D-3008 Garbsen
Atari ST Club Hannover
Hermann-Löns-Straße 40
D-3008 Garbsen 4

D-3200 Hildesheim
ST Anwender Club
Horst Kraus
Trockener Kamp 29
D-3200 Hildesheim

D-3450 Holzminden
CIG ASTAG
Comp. Interessen Gemeinschaft
Postfach 14 09
D-3450 Holzminden 1

D-4040 Ratingen
Basta
Postach 15 45
D-4030 Ratingen 1

D-4030 Ratingen
VDA
Verein der Atari PC Anwender e.V.
Straßburger Str. 5
D-4030 Ratingen 1

D-4250 Bottrop
Atari ST User Club
Andreas Czech
In der Schanze 20
D-4250 Bottrop

D-4408 Dülmen
Dülmener Atari ST Computer Club
Dietmar Rabich
Dövelingsweg 2
D-4408 Dülmen

D-4600 Dortmund
STUDO
ST-User Dortmund
J. Finkemeier
Kreuzstr. 65
D-4600 Dortmund

D-4800 Bielefeld
Microtec
Carl-Severing-Str. 190
D-4800 Bielefeld 14

D-4970 Bad Oeynhausen
CCC-SVHI-Minden
D-4970 Bad Oeynhausen

D-5276 Wiehl
ST-Club Oberberg
Thomas Höser
Siefenerstr. 10
D-5276 Wiehl

D-5300 Bonn
ST-User Club Bonn
Alexander Görbing
Siegstr. 12
D-5300 Bonn

D-6090 Rüsselsheim
STAC
Postfach 12 25
D-6090 Rüsselsheim

D-6070 Langen
ST VISION
Kai-Uwe Wahl
Postfach 16 51
D-6070 Langen

D-6097 Trebur
RHEINMAIN-CLUB-ATARI
Eckard Linner
Am Mittelpfad 33
D-6079 Trebur 2

D-6113 Babenhausen
CCC
W. Hofmann
Altdörferstr. 23
D-6113 Babenhausen

D-6220 Geisenheim
Interessengruppe für Bild- und Schriftübertragung
Karl-Heinz Geiß
Waldstr. 9
D-6220 Geisenheim

D-6419 Eiterfeld
ATARI Computer Club Fulda
Rüdiger Weis
Antsanvia 5
D-6419 Eiterfeld

D-6477 Limeshain
ATARI-User-Club Limeshain
Christoph Lier
Langestraße 11
D-6477 Limeshain

D-6500 Mainz
STURM
ST User Rhein Main
Jürgen Wirth
Weinstr. 17
D-6500 Mainz

D-6520 Worms
UTOPIA
Volkhard Werner
Alter Kirchenpfad 8
D-6520 Worms 26

D-6650 Homburg
HT – lg/kb.
Vorgartenstr. 9
D-6650 Homburg 8

D-6710 Frankenthal
User Club ST
Her Jung
Zuckerfabrikstr. 3
D-6710 Frankenthal

D-7000 Stuttgart
ACS-ATARI Club Stuttgart
Hans-Martin Bregler
Duisburgerstr. 29/3
D-7000 Stuttgart 50

D-7050 Waiblingen
ATARI ST Rems Murr
Jürgen Wegeer
Karlstr. 32
D-7050 Waiblingen

D-7100 Heilbronn
CIP
Club für Informatik + Programmierung
Ulrich Veigel
Köhlstraße 9
D-7100 Heilbronn

D-7120 Heilbronn
Computer Treff
im Bürgerhaus Böckingen
Kirchsteige 5
D-7120 Heilbronn Böckingen

D-7400 Tübingen
AKT-Club
Hirschauerstr. 18
D-7400 Tübingen

D-7520 Bruchsal
CUVB
Computer User Vereinigung
Bruchsal e.V.
Kaiserstr. 16
D-7520 Bruchsal

D-7600 Offenburg
Atari ST Anwender Club
J. Ruschmann
Im Heldenrecht 9
D-7600 Offenburg

D-7800 Freiburg
CDSt
Werner Laass
Haslenweg 3
D-7800 Freiburg

D-7920 Heidenheim
A.P.D.C.G.
c/o Frank Reinhard
Carl-Spitzweg-Straße 17
D-7920 Heidenheim 5

D-8110 Murnau
ATARI ST-Club Oberbayern
Postfach 13 15
D-8110 Murnau

D-8201 Grosskarolinenfeld
ATARI ST Interessengruppe
Ernst Huber
Rosenheimerstraße 22
D-8201 Grosskarolinenfeld

D-8671 Schönwald
ST Computerclub MEGA
Michael Sommerer
Marienstr. 18
D-8671 Schönwald

D-7080 Gerbrunn
ATARI Club
Reinhard Full
Nikolaushöhe 1
D-8700 Würzburg

A-1011 Wien
CLUB 68000
Postfach 320
A-1011 Wien

A-1150 Wien
ST Gesprächsrunde Wien
Franz G. Szabo
Sechshauserstr. 59/3/19
A-1150 Wien

A-6020 Innsbruck
ATARI ST Interessengruppe
Gerald Gunsch
Arzler Straße 122
A-6020 Innsbruck

A-6176 Völs
G.E.M.-Computerclub Innsbruck
Norbert Kreyer
Wolkensteinstr. 4b
A-6176 Völs

A-6300 Wörgl
ATARI ST Interessengruppe
Curt Zimmermann
Innsbruckerstraße 12
A-6300 Wörgl

A-6330 Kufstein
ATARI ST Interessengruppe
Manfred Recla
Terlanerstraße 8
A-6330 Kufstein

A-8010 Graz
ST Gesprächsrunde Graz
Günther Jernej
Rosenhaingasse 7
A-8010 Graz

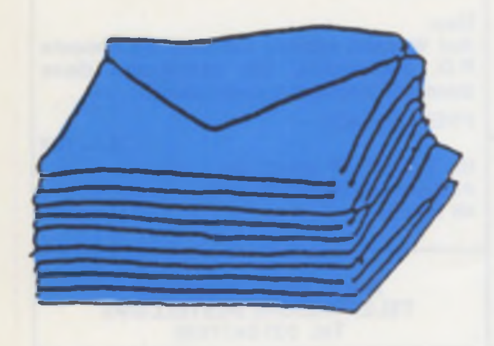
CH-3047 Bremgarten
STAB
ST Anwender Bern
Theodor Mohr
Zelweg 3
CH-3047 Bremgarten

CH-6600 Locarno
User Club Ticino
c/o Norberto R. Wittwer
Via Alfredio Pioda 1
CH-6600 Locarno

B-4700 Seraing
Club EMC ATARI ST
c/o Christian Poels
Rue des Bas-Sarts, 10
B-4100 Seraing-Belgipue

DK-8000 Aarhus C
Midtjyllands 68000 ST User Group
c/o Andreas Ramos
Jaegerguardsgade 142.2
DK-8000 Aarhus C

Leserecke & Leserbriefe



Ich habe ein Problem, welches meine DFÜ-Arbeit außerordentlich stört: Ohne eigene Tastatureingabe oder spontan während einer beliebigen Eingabe tauchen „Geisterzeichen", meist in Form von „geschweiften Klammern" mit nachfolgendem „i" auf dem Bildschirm auf, welche im Zusammenhang mit meiner Eingabe natürlich vom Mailboxcomputer nicht akzeptiert wird. Das Löschen dieser Zeichen mittels „Backspace" und das oft mehrfache Wiederholen der Eingabe ist lästig, zeitraubend und kostenintensiv. Die Störung tritt sowohl über „Normalleitung" wie über Datex-P auf, in verschiedenen Mailboxen, und, soweit ich mich erinnere auch mit verschiedenen Terminalprogrammen.
Ich arbeite hauptsächlich mit PC/Intercomm in der Konfiguration: 1200 Bd., 8N1, Vollduplex, Echo an.' Eine andere Störung, die mindestens ebenso entnervend ist, taucht während des Empfangs von längeren ASCII-Texten auf: Ganz plötzlich gibt der Bildschirm über viele Zeilen ein kariertes Quadrat aus, welches für ein undefiniertes Zeichen steht. Nach längerem warten, bei dem natürlich eine Menge Informationen verloren gehen, setzt der Textfluß gelegentlich spontan wieder ein, sofortige Abhilfe schafft allerdings das Betätigen von <CONTR><S>, gefolgt von <COLNTR><Q>, so, als wenn hierdurch wieder eine Synchronisierung der Daten mit Erkennung von Start-Stopp-Bit erfolgen würde.
Meine Frage lautet nun:

1.) Tauchen diese Übertragungsfehler auch bei anderen Teilnehmern auf?
2.) Handelt es sich hier um akustische Leitungsstörungen, die zufällig im Frequenzbereich der Koppler/Modems liegen, und die vom Terminalprogramm als (nicht definiertes) Zeichen interpretiert wird, woraufhin es allerdings die Synchronisation zu den nachfolgenden korrekten Zeichen verliert und seitenweise Zeichensalat ausgibt?
3.) Gibt es eine Einstellung z. B. bei PC/Intercomm oder bei einem anderen komfortablen Terminalprogramm, welches diese Übertragungsfehler kompensiert?

Ulrich Keil, Düsseldorf

Ich besitze einen Atari 520 ST+ und seit kurzem auch einen Brother M-1409 Drucker. Nun habe ich bei der Hardcopy folgendes Problem:
Bei dem ausgedruckten Bild entsteht auf der rechten Seite ein etwa 4 cm breiter Streifen, der etwas heller gedruckt wird. Auf diesem Streifen sind ganz dünn Teile des linken Bildrandes, jeweils um eine Zeile verrückt, zu erkennen.
Wer kann mir sagen woran das liegt und wie man diesen Fehler beheben kann?

Kai-Immo Wels, Bad Oldesloe

Zunächst einmal ein paar Sätze in eigener Sache: Bitte geben Sie bei Ihren Leserbriefen unbedingt Ihre Telefonnummer mit an. Viele Probleme lassen sich direkt am Telefon mit ein paar Worten beseitigen und Sie ersparen sich eventuelle Wartezeiten. Danke!

Betr.: Leserbrief von Herrn Dr. L. Pape aus der ST 9/87
Hardcopy mit OKI ML 192

Hardcopies lassen sich mit dem Programm MULTI-HARDCOPY, das von diversen Händlern (z. B. ST 9/87 S. 71) für DM 98,– angeboten wird, erstellen. Eine kostengünstigere Lösung gibt's mit dem Programm T-Hardfix.PRG von Torsten Anders, erschienen im ST-Magazin Sonderheft der Happy Computer.

H. D. Schultz, Ludwigsburg

Betr.: Leserbrief von Herrn Rene Lieb aus der ST 9/87
VIP (Lotus 1-2-3) Profi's

Ja, ja, Eingabemasken und der Transpose-Befehl. VIP entspricht leider nur der LOTUS-Version 1.A, also einer der ersten LOTUS-Versionen überhaupt. Inzwischen gibt es von LOTUS Development eine Version 2.01, die einen erheblichen erweiterten Befehls-, Macro- und Formel-Sprachsatz besitzt. (Sie verwenden wohl diese Version). leider werden die ATARI-VIP's vergeblich auf eine VIP V.2.0 mit dem Transpose-Befehl warten müssen. Die Programmentwickler denken nicht an eine Weiterentwicklung, wohl aber an eine intensive Käferjagd. Doch Ihre Eingaben-Problematik ohne 'Transpose' läßt sich auch mit einem Macro oder der /x1 und /xn in Kombination mit der Copy-Funktion lösen. Wenn Sie nicht weiterkommen, schicken Sie mir Ihr Template auf Disk (5 1/4" oder 3 1/2"). Vielleicht gibt es eine einfache Super-Macro-Lösung. Rückporto DM 2,50 nicht vergessen. Heinz D. Schultz, Ingersheimer Str. 12, 7140 Ludwigsburg.

Betr.: Leserbrief von St. Bauer aus der ST 9/87 – Fehlerhaftes Accessory auf der Festplatte

Auch mir ist dasselbe passiert – ein fehlerhaftes Accessory auf Partition C der Festplatte – ergab ein Endlos-Reset. Was tun? Da ich keine Diskette mit dem TOS besitze, mußte ich auf eine andere Lösung kommen: Start des Systems ohne die Festplatte einzuschalten (mit einer Diskette, die nur die Harddisk C anmeldet) – Festplatte einschalten und Treiber-Programm (AHDI.PRG) aus dem Auto-Ordner der Bootdiskette vom Desktop aus starten – fehlerhaftes Accessory löschen – Reset (mit normaler Harddisktreiber-Diskette) – alles ok! Dies als Hilfe für diejenigen, denen das schon passiert ist oder noch passieren wird.

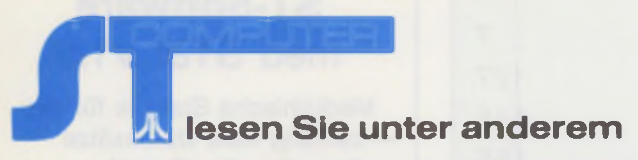
Roland Ebner, CH-Möhlin

Betr.: Artikel „Im übertragenen Sinne", ST Computer 7/8/87

In einer Ihrer vorigen Ausgaben hatten Sie ein Interface, bestehend aus Hard- und Software zur Kopplung des ATARI ST mit einem SHARP Taschenrechner 1401/2 vorgestellt. Leider habe ich die entsprechende Ausgabe nicht mehr wiedergefunden.
Die Adresse lautet:
ECPS Rainer Kratzer
Zur Ziegelhütte 6
6992 Adelsheim-Leibenstadt
Tel.: 06291/7731

Inserentenverzeichnis

In der nächsten



lesen Sie unter anderem



Das neue SIGNUM!

Endlich beherrscht SIGNUM! auch die Grafik. Das Einbinden von Grafiken in den Text ist jetzt möglich. Doch nicht nur dieser wichtige Punkt ist hinzugekommen. Spaltensatz und vieles mehr, lassen der Kreativität des Anwenders freien Lauf.

Die Zeit der Messen

Wenn die ersten Blätter sich zu verfärben beginnen, ist es meist wieder Messezeit. Wir waren für Sie auf der **Systems 87** in München und auf der **PCW 87** in London. Was es neues auf dem Soft- und Hardwarebereich für den ST gibt, können Sie in der nächsten Ausgabe lesen. Man darf gespannt sein.

Der schnelle Zugriff

– zwei neue Festplatten

Die neue ATARI Festplatte SH 205 ist da. Was sie leistet und was sich gegenüber der alten SH 204 geändert hat, kann man in der Dezember-Ausgabe erfahren.

Die zweite Harddisk, die wir vorstellen wollen, ist eigentlich ein Aufrüstungsset für 20 MB Festplatten. Durch dieses Set kann man die Kapazität seiner eigenen Platte problemlos vergrößern.

Fibuman

– ein neues professionelles Finanzbuchhaltungspaket

Mit Fibuman ist zur Zeit ein modular aufgebautes Finanzbuchhaltungspaket für die ATARI ST Serie auf dem Markt. Wir stellen es in seiner neuesten Version vor.

Die ST-Computer Ausgabe 12/87 erscheint am 27.11.1987

Impressum

ST-Computer

Verlag:
Heim Fachverlag,
Heidelberger Landstraße 194,
6100 Darmstadt 13,
Telefon (0 61 51) 5 60 57 FAX 0 61 51 / 5 56 89

Verlagsleitung:
Hans-Jörg Heim

Redaktion:
'Merlin' Computer GmbH
Uwe Bärtels (UB Chefredakteur)
Marcelo Merino (MM)
Harald Egel (HE)
Harald Schneider (HS)

Anschrift:
'Merlin' Computer GmbH
ST-Computer Redaktion
Postfach 59 69
Industriestr. 26
6236 Eschborn
Tel. 0 61 96 / 48 18 11 FAX 0 61 96 / 4 11 37

Redaktionelle Mitarbeiter:
Markus Nerding (MN), Uli Eickmann (UE), Jürgen Leonhard (JL), Jörg Wilhelm (JW), C. P. Lippert, R. Hofmann, Oliver Joppich (OJ), Stefan Höhn (SH), Chr. Schormann (CS), C. Brod, A. Suchy

Kommunikation und Presseleitung:
Claus P. Lippert

Autoren dieser Ausgabe:
M. Koller, R. Hauke, Dr. Sarnow, D. Beyer, I. Brümmer, S. Gleissner, C. Hoofe, P. Lerailler, P. Sadoni

Produktion:
Klaus Schultheis (Ltg.),
Patricia Illing
Bernd Uwe Failer
Susanne Failer
Bela Kumar
Ilka Lindemann

Anzeigenverkaufsleitung:
Uwe Heim

Anzeigenpreise:
nach Preisliste Nr. 2, gültig ab 1.7.86

Vertrieb:
Hans-Jörg Heim, Uwe Heim,
Heide Schultheis

Erscheinungsweise:
11 x jährlich

Bezugspreis:
Einzelheft DM 6,–
Jahresabonnement DM 60,– inklusive der gesetzlichen Mehrwertsteuer und den Zustellgebühren für 11 Ausgaben.
Ausland: 80,– DM inkl. Versand

Bezugsmöglichkeiten:
ATARI-Fachhändler, Zeitschriftenhandel, Kauf- und Warenhäuser oder direkt beim Verlag unter obiger Adresse.

Druck:
Ferling Druck Darmstadt

Manuskripteinsendungen:
Programmlistings, Bauanleitungen und Manuskripte werden von der Redaktion gerne angenommen. Sie müssen frei von Rechten Dritter sein. Mit ihrer Einsendung gibt der Verfasser die Zustimmung zum Abdruck und der Vervielfältigung auf Datenträgern im Heim Verlag. Honorare nach Vereinbarung. Für unverlangt eingesandte Manuskripte wird keine Haftung übernommen.

Titelseite: Fabian & Mayer
Foto: Rainer Spirandelli

ISSN 0932-0385

Für alle ATARI ST

neu von GFA

Animations-Programm für ATARI ST

ATARI ST

GFA-MOVIE

GFA MOVIE

5teiliges, unkompliziertes Programmpaket
- Ansprechendes Zeichenprogramm
- Vielfältiger Zeichensatzeditor
- Leistungsfähiges Drehbuchprogramm
- Vielseitiges Effektprogramm
- SnapShot-Möglichkeit

Erstellte Filme laufen eigenständig
Extrem schnell und vollständig in deutsch
Komplett mit der Maus zu bedienen
Benötigt monochromen Monitor

GFA-MOVIE DM 149,–

Desktop Publishing Programm für ATARI ST

ATARI ST

GFA-PUBLISHER

GFA PUBLISHER

Voll GEM-gesteuertes Desktop Publishing Programm mit leicht erlernbarer Bedienerführung
Interne Rasterauflösung von 2540 Punkten pro Zoll
Bemaßung in mm, Zoll, Didot-Punkten, Cicero, Points und Pica
Integration der gängigen ATARI ST Grafikformate sowie GEM Image-Format und Metafile-Format
Wandlung der Farbbildformate in Halbtonvorlagen
Spezielle Funktionen zur Manipulation von Halbtonvorlagen
Erzeugung und Einzug von Bildern im Bitformat
Leistungsfähiger Texteditor
Automatischer Textfluß und -umbruch
Leistungsfähige dreisprachige Silbentrennung: deutsch, französisch und englisch (sowohl automatisch als auch manuell)
Erweiterbare Schriftenbibliothek
Integrierte Treiber für POSTSCRIPT* und HP-GL*
Weitere Druckertreiber nachladbar

Desktop Publishing Programm

GFA-PUBLISHER
DM 398,–

Design-Programm für ATARI ST

ATARI ST

GFA-ARTIST

GFA ARTIST

- Alle erdenklichen Gestaltungsmöglichkeiten von Farbgrafik und deren Animation
- Erstellen eines Films durch Zusammensetzen verschiedener Animationen
- Arbeitet mit bis zu 1024 Farben gleichzeitig (auch colorcycling)
- Gemeinsame Darstellung der vollen Farbpalette sowie 80-Zeichen-Text
- Kompatibel zu Standard-Malprogrammen
- Erstellte Filme laufen eigenständig
- Extrem schnell und vollständig in deutsch

Benötigt ATARI ST mit 1 MB ROM + ROM-TOS + Farbmonitor

GFA-ARTIST DM 149,–

...Anruf genügt: 02 11/58 80 11
GFA-CLUB, GFA-PC-Software bitte Info anfordern

GFA Systemtechnik GmbH

Heerdter Sandberg 30
D-4000 Düsseldorf 11
Telefon 02 11/58 80 11

GFA SYSTEMTECHNIK